

UNIVERSITÉ DE SHERBROOKE  
Faculté des sciences appliquées  
Département de génie électrique et génie informatique

**Identification de la non-linéarité d'un système de contrôle de  
vibration par des modèles basés sur les réseaux de neurones**

Mémoire de maîtrise ès sciences appliquées  
Spécialité : génie électrique

Yasdeo BISSESSUR

Sherbrooke (Québec), CANADA

Juin 1997

## RÉSUMÉ

De nos jours, les technologies d'actuateurs et de capteurs piézo-électriques offrent des solutions intéressantes dans le contrôle actif des vibrations. Pour ce type d'actuateur et de capteur, l'efficacité d'un contrôle actif est limitée par la non-linéarité de l'arrangement actuateur/colle/système mécanique/colle/capteur. L'identification de cet arrangement est une étape très importante dans le développement des contrôleurs. Pour tenir compte de la non-linéarité du système dans l'identification, des modèles non-linéaires sont requis.

Dans ce projet, on commence par utiliser des excitations en régime harmonique pour évaluer l'importance quantitative et le comportement général de la non-linéarité. Ensuite, on évalue les performances des identifications basées sur des modèles linéaires. Finalement, on fait des identifications du système par des modèles non-linéaires basés sur les réseaux de neurones. D'après les expérimentations faites, ces derniers sont des meilleurs candidats que les modèles linéaires, pour l'identification de la branche de contrôle.

## REMERCIEMENTS

Je dois beaucoup à mon directeur de recherche, Professeur Bruno Paillard, pour m'avoir accordé ce projet, pour ses nombreux encouragements, son support moral et technique, et pour la confiance qu'il m'ait faite durant toute la durée de ces travaux de recherche. Je voudrais aussi le remercier pour un très bon cours de contrôle actif adaptatif qui m'a donné un bon avancement à mes recherches.

Je suis autant reconnaissant au Professeur Chon Tam LeDinh qui était toujours prêt à faire part des nombreuses suggestions, pour ses précieux conseils, pour son support moral et sans qui ce projet n'aurait pas aboutit. En effet, son expertise avec les réseaux de neurones a été un atout indispensable à la réussite de ce projet.

Je voudrais remercier Martin Bouchard, Philippe Micheau, et Patrice Masson qui étaient toujours prêts à discuter des nombreux problèmes et pour leurs aides bibliographiques. Mes remerciements vont aussi à Suzanne St-Laurent, pour avoir lu le mémoire et corrigé les nombreuses fautes de français.

Finalement, je suis très reconnaissant au ministère de l'éducation du gouvernement du Québec, sans lequel mes études de deuxième cycle à l'Université de Sherbrooke n'auraient pas été envisageables.

## TABLE DES MATIÈRES

	Page
1. INTRODUCTION.....	1
2. NON-LINÉARITÉ DU SYSTÈME ET MÉTHODES D'IDENTIFICATIONS.....	7
2.1 Introduction.....	7
2.2 Non-linéarité de la branche de contrôle.....	7
2.3 Distortion harmonique et fréquence d'échantillonnage.....	8
2.4 Identification des systèmes.....	9
2.5 Identification par filtrage adaptatif linéaire.....	10
2.6 Identification par séries de Volterra.....	11
2.7 Réseaux de neurones de type feedforward.....	14
2.8 Identification par réseaux de neurones.....	18
2.9 Bruit dans le système physique.....	25
2.10 Conclusion.....	26
3. ÉVALUATION DE LA NON-LINÉARITÉ GLOBALE PAR EXCITATION EN RÉGIME HARMONIQUE.....	28
3.1 Introduction.....	28
3.2 Procédure de l'expérimentation.....	28
3.3 Résultats.....	30
3.4 Conclusion.....	36
4. IDENTIFICATIONS DU SYSTÈME PAR MODÈLES LINÉAIRES.....	37
4.1 Introduction.....	37
4.2 Algorithme LMS pour filtre linéaire.....	37
4.3 Identification du système par modèle MA adaptatif.....	40
4.4 Identification des filtres d'acquisitions par modèle MA.....	42
4.5 Identification du système par modèle ARMA adaptatif.....	45
4.6 Conclusion.....	52
5. IDENTIFICATION NON-LINÉAIRE PAR RÉSEAUX DE NEURONES.....	53
5.1 Introduction.....	53
5.2 Modèles d'identification non-linéaires.....	53
5.3 Réseau de neurones feedforward de deux couches.....	54
5.4 Algorithme de rétropropagation améliorée.....	57
5.5 Modification sur l'algorithme de rétropropagation améliorée....	60



5.6	Méthodes pour les identifications.....	65
5.7	Méthode de vérification des modèles identifiés.....	71
6.	RÉSULTATS DES IDENTIFICATIONS NON-LINÉAIRES...	73
6.1	Introduction.....	73
6.2	Identification basée sur le modèle I.....	73
6.3	Identification basée sur le modèle II.....	75
6.4	Identification basée sur le modèle III.....	77
6.5	Identification basée sur le modèle IV.....	78
6.6	Vérification des modèles identifiés pour excitations en bruit blanc	80
6.7	Vérification des modèles pour excitations en régime harmonique	83
6.8	Optimisation du réseau pour le model IV.....	90
6.9	Commentaires sur les résultats.....	92
	CONCLUSION.....	94
	Annexe 1: Modes de vibration pour une poutre uniforme encastrée-libre	96
	Annexe 2: Modélisation d'un capteur PVDF.....	97
	BIBLIOGRAPHIE.....	98

## LISTE DES FIGURES

	Page
Figure 1.1 Arrangement pour étude de contrôle actif de vibration	1
Figure 1.2 Dimensions de la poutre	2
Figure 1.3 Contrôle par anticipation	4
Figure 1.4 Contrôle par rétroaction	4
Figure 2.1 Identification par filtre transversal adaptatif	10
Figure 2.2 Filtre Volterra adaptatif	13
Figure 2.3 Réseau feedforward de deux couches	15
Figure 2.4(a) Modèle I	19
Figure 2.4(b) Modèle II	20
Figure 2.4(c) Modèle III	20
Figure 2.4(d) Modèle IV	21
Figure 2.5 Identification en parallèle	22
Figure 2.6 Identification série-parallèle	23
Figure 2.7(a) Bruit à la sortie avec l'actuateur en circuit ouvert	25
Figure 2.7(b) Spectre d'énergie du bruit	26
Figure 3.1 Arrangement pour évaluer la distortion harmonique	29
Figure 3.2(a) Spectre du signal de sortie pour une excitation de 3.0 Hz	30
Figure 3.2(b) Spectre du signal de sortie pour une excitation de 5.0 Hz	31
Figure 3.2(c) Spectre du signal de sortie pour une excitation de 7.5 Hz	31
Figure 3.2(d) Spectre du signal de sortie pour une excitation de 17.0 Hz	32
Figure 3.3 Vue de haut de la poutre	33
Figure 3.4 Fonction de transfert du système	34
Figure 4.1 Identification par filtre linéaire transversal	37
Figure 4.2 Réponse impulsionnelle identifiée	40
Figure 4.3 Fonction de transfert identifiée	41

Figure 4.4	Réponse impulsionnelle de l'arrangement A/N-N/A en série	42
Figure 4.5	Fonction de transfert de l'arrangement A/N-N/A en série	43
Figure 4.6	Fonction de transfert de chaque convertisseur	44
Figure 4.7	Réponse impulsionnelle de chaque convertisseur	44
Figure 4.8	Identification d'un modèle ARMA par la méthode de l'erreur d'équation	46
Figure 4.9	Coefficients $\{b_k\}$ pour le modèle ARMA identifié	48
Figure 4.10	Spectres du signal cible et d'erreur et fonction de transfert identifiée	48
Figure 4.11	Réponse impulsionnelle calculée	49
Figure 4.12	Réponse impulsionnelle mesurée	50
Figure 4.13	Différence entre les réponses	50
Figure 4.14(a)	Agrandissement sur les premiers 100 échantillons	51
Figure 4.14(b)	Agrandissement sur les échantillons entre 300 et 450	51
Figure 5.1	Réseau de neurones de deux couches	55
Figure 5.2	Identification par réseau de neurones	57
Figure 5.3	Structure pour l'identification basée sur le modèle I	67
Figure 5.4	Structure pour l'identification basée sur le modèle II	68
Figure 5.5	Structure pour l'identification basée sur le modèle III	69
Figure 5.6	Structure pour l'identification basée sur le modèle IV	70
Figure 5.7	Vérification du modèle III en mode récursif	72
Figure 6.1	Un bloc du signal cible	74
Figure 6.2	Erreur à la fin de l'identification (modèle I)	74
Figure 6.3	Spectres des signaux de cible et d'erreur (modèle I)	75
Figure 6.4	Erreur à la fin de l'identification (modèle II)	76
Figure 6.5	Spectres des signaux de cible et d'erreur (modèle II)	76

Figure 6.6	Erreur après l'adaptation du modèle III	77
Figure 6.7	Spectres des signaux de cible et d'erreur (modèle III)	78
Figure 6.8	Erreur d'identification pour le dernier bloc de signal (modèle IV)	79
Figure 6.6	Spectres des signaux de cible et d'erreur pour modèle IV	79
Figure 6.10(a)	Réponse du modèle III en comparaison avec celle du système	80
Figure 6.10(b)	Erreur de modélisation (modèle III)	80
Figure 6.10(c)	Spectres des réponses du système et du modèle III	81
Figure 6.11(a)	Réponse du modèle IV en comparaison avec celle du système	81
Figure 6.11(b)	Erreur de modélisation (modèle IV)	81
Figure 6.11(c)	Spectres des réponses du système et du modèle IV	82
Figure 6.12(a)	Vérification du modèle III : excitation à 3.0 Hz	83
Figure 6.12(b)	Vérification du modèle III : excitation à 4.0 Hz	84
Figure 6.12(c)	Vérification du modèle III : excitation à 7.5 Hz	84
Figure 6.12(d)	Vérification du modèle III : excitation à 9.0 Hz	85
Figure 6.12(e)	Vérification du modèle III : excitation à 12.0 Hz	85
Figure 6.12(f)	Vérification du modèle III : excitation à 17.0 Hz	86
Figure 6.13(a)	Vérification du modèle IV : excitation à 3.0 Hz	86
Figure 6.13(b)	Vérification du modèle IV : excitation à 4.0 Hz	87
Figure 6.13(c)	Vérification du modèle IV : excitation à 7.5 Hz	87
Figure 6.13(d)	Vérification du modèle IV : excitation à 9.0 Hz	88
Figure 6.13(e)	Vérification du modèle IV : excitation à 12.0 Hz	88
Figure 6.13(f)	Vérification du modèle IV : excitation à 17.0 Hz	89

## LISTE DES TABLEAUX

Page

TABLEAU 3.1	AMPLITUDES DES HARMONIQUES DANS LE SIGNAL DE SORTIE POUR EXCITATION EN SINUS DE DIVERSES FRÉQUENCES	33
TABLEAU 3.2	NIVEAUX DES HARMONIQUES APRÈS LA COMPENSATION DE L'EFFET DE LA FONCTION DE TRANSFERT ET NORMALISATION	35
TABLEAU 6.1	EFFET DE VARIER $N_x$ ET $N_y$ SUR LA CONVERGENCE	91
TABLEAU 6.2	EFFET DE VARIER $N_1$ POUR $N_x = 32$ ET $N_y = 24$	91

*A mes chers parents*

*et*

*à tous ceux et celles qui m'ont inspiré*

## 1. INTRODUCTION

Aujourd'hui, de nombreuses technologies d'actuateurs et de capteurs sont utilisées en contrôle actif du bruit et des vibrations. Parmi ces technologies les actuateurs et capteurs piézo-électriques offrent des solutions intéressantes en raison de leur compacité et de leur légèreté. Ils peuvent en effet être étendus (collés) en couches minces sur les structures mécaniques à contrôler, ne modifiant pas de façon importante la structure.

Un arrangement très simple pour étudier le contrôle actif de vibration sur une poutre encastrée-libre par cette technologie est illustré dans la figure 1.1. Cet arrangement existe déjà sur des maquettes utilisées pour le développement des techniques de contrôle actif de vibration à l'université de Sherbrooke.

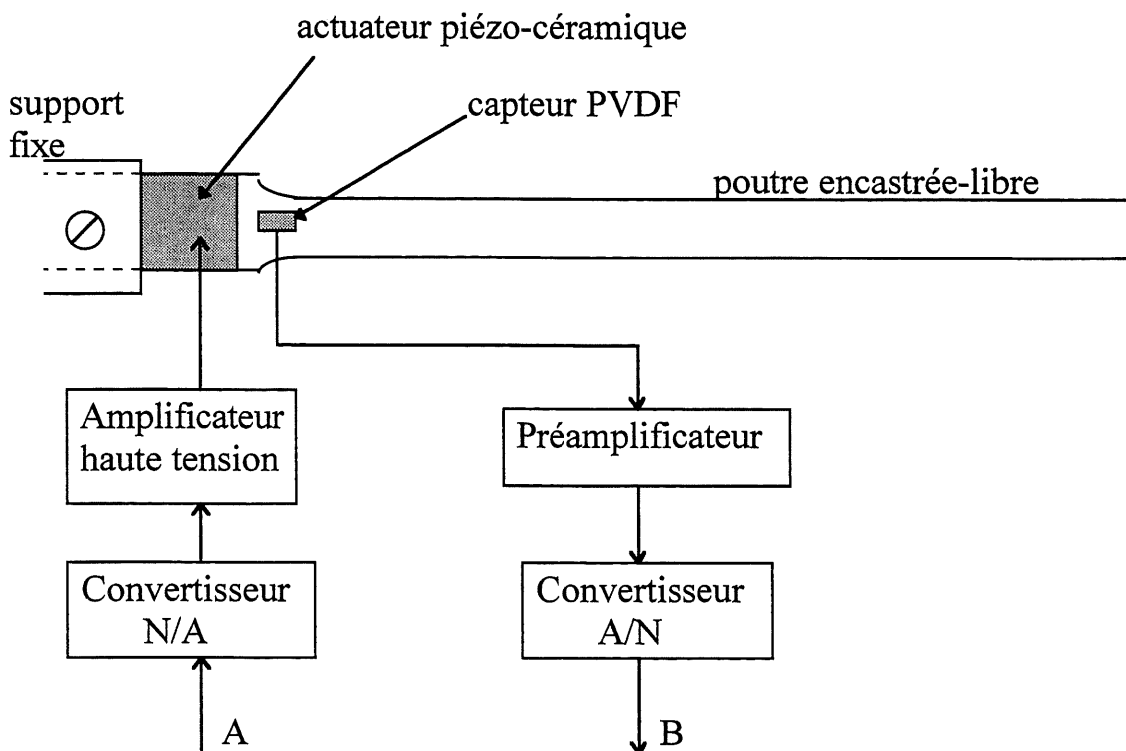


Figure 1.1 Arrangement pour étude de contrôle actif de vibration

Les dimensions de la poutre sont données dans la figure 1.2. La première fréquence de résonance de la poutre est très proche de 15.0 Hz. Cette valeur est obtenue en observant la fonction de transfert du système, lors des identifications basées sur des modèles linéaires (voir le chapitre 4). Si on essaye de calculer la première fréquence propre de la poutre avec la formule donnée dans l'annexe 1, en utilisant les dimensions données dans la figure 1.2 ( $L = 367$  mm,  $h = 2.00$  mm), une densité de  $2710 \text{ kg/m}^3$  et un module de Young de  $6.7 \times 10^{10}$  Pa pour l'aluminium, on obtient une valeur de 11.9 Hz. Cette différence est en partie due au fait que l'actuateur piézo-céramique collé sur la poutre modifie légèrement la structure (l'épaisseur augmente légèrement et les valeurs effectives pour le module de Young et la densité sont aussi altérées). Les autres facteurs qui pourraient être responsables de cette différence sont la non-uniformité de la poutre et l'imprécision sur la valeur du module de Young utilisée. La deuxième fréquence propre de la poutre est calculée à 74.6 Hz en utilisant la formule de l'annexe 1. Pour les mêmes raisons, on pourrait s'attendre que la deuxième fréquence de résonance de la poutre soit un peu plus élevée que la valeur calculée.

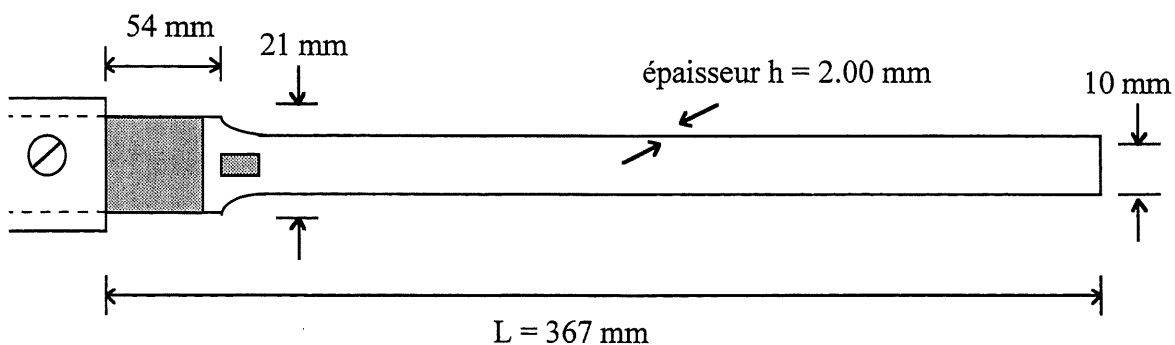


Figure 1.2 Dimensions de la poutre



L'actuateur est formé de deux tranches piézo-électriques collées de chaque côtés de la poutre (en aluminium). En appliquant une tension assez haute sur l'actuateur, l'une des cellules piézos se contracte, tandis que l'autre s'étire dans la direction de la longueur de la poutre. Cela résulte en un moment de flexion qui fait fléchir la poutre latéralement.

Une déflexion de la poutre produit une charge sur le capteur PVDF proportionnelle à la déflexion. Cela résulte en une tension qui apparaît à l'entrée du préamplificateur qui l'amplifie. Le convertisseur analogique-numérique transforme la sortie du préamplificateur en un signal numérique.

Les points A et B dans la figure 1.1 correspondent à des points spécifiques sur une carte DSP (appelée DSKX) pour les signaux entrée et sortie du système. Tout le système entre les points A et B, du point de vue de la fonction de transfert, est référé comme la branche de contrôle et il est représenté par C dans ce qui suit.

Dans une situation de contrôle de vibration, la poutre encastrée-libre est soumise à une perturbation externe pouvant être due à un couplage mécanique au support ou à une autre force appliquée sur la structure. On dit qu'une onde de perturbation existe sur la structure. Le principe de contrôle est d'appliquer une anti-onde sur la structure pour annuler la vibration. L'anti-onde est appliquée via l'actuateur piézo-électrique et elle provient d'un contrôleur. Le signal de sortie du contrôleur est référé comme le signal de contrôle. L'effet de la combinaison de l'onde et de l'anti-onde est mesuré par le capteur PVDF et un signal d'erreur de contrôle est disponible au point B.

Deux configurations sont possibles pour le contrôle : contrôle par anticipation (*feedforward control*) et contrôle par rétroaction (*feedback control*). La première est illustrée dans la figure 1.3. Dans cette configuration, un signal de référence, provenant d'un capteur à la source de la perturbation, est disponible et il est utilisé comme référence pour le contrôleur. La fonction de transfert entre ce capteur et le point B est représentée par P.

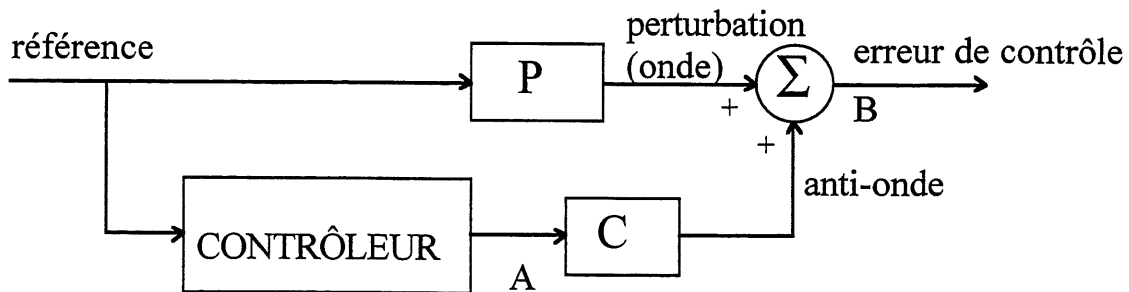


Figure 1.3 Contrôle par anticipation

Dans le cas où il n'est pas possible d'avoir une référence à la source de la perturbation, un contrôle par rétroaction est plus pertinent. Cette configuration est illustrée dans la figure 1.4. Remarquons que le signal d'entrée du contrôleur provient du capteur PVDF.

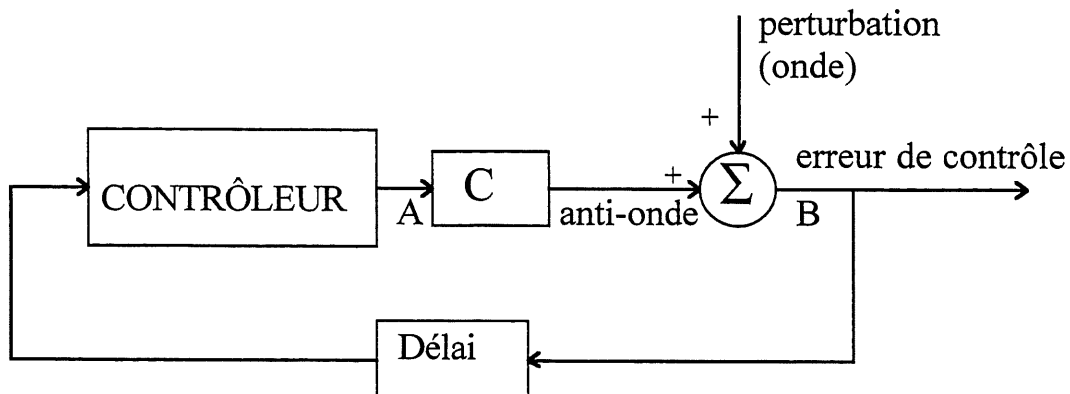


Figure 1.4 Contrôle par rétroaction

Dans ces deux configurations, le problème est le développement et la conception d'un contrôleur optimal. Pour ce faire, il est très important d'avoir une représentation précise de la branche de contrôle C. Le processus permettant d'obtenir un modèle numérique pour représenter un système physique, en ce qui concerne les signaux d'entrée et de sortie, est appelé une identification.

Pour une technologie utilisant des actionneurs et capteurs piézo-électriques, la non-linéarité de la branche de contrôle pose des problèmes pour un contrôle optimal. Pour contourner ces problèmes, des contrôleurs non-linéaires sont requis. Il serait alors très important d'avoir un modèle non-linéaire pour la branche de contrôle pour tenir compte de sa non-linéarité.

L'objectif de ces recherches est d'identifier un modèle non-linéaire qui serait plus capable de représenter la branche de contrôle que le ferait un modèle linéaire. Une identification basée sur un modèle par moyenne ajustée (*moving average process* ou MA) donne une erreur d'identification avec une énergie d'environ -20 dB en moyenne par rapport à l'énergie du signal de sortie, c'est-à-dire que la différence entre la sortie du système physique et celle du modèle identifié est en moyenne 10% de l'amplitude moyenne de la sortie du système physique.

Dans ce document, on présente les travaux de recherches qui ont été faits à ce sujet. On commence par exposer l'état de l'art en ce qui concerne l'identification des systèmes non-linéaires. Ensuite, une évaluation globale de la non-linéarité basée sur des excitations en harmoniques est faite. On procède avec des identifications basées sur des modèles linéaires pour évaluer leurs

performances. Finalement, on considère des identifications par des modèles basés sur les réseaux de neurones.

Une grande quantité de littérature est disponible à ce sujet. L'article [20] a été important pour établir la fréquence d'échantillonnage des signaux analogiques. Pour les algorithmes de filtrage adaptatif, l'ouvrage [5] est très utile. La littérature sur les réseaux de neurones et les algorithmes d'apprentissages provient principalement de [2, 7, 8, 10, 15]. L'utilisation des réseaux de neurones dans le contrôle actif est documentée dans [1, 3, 9, 10].

## 2. NON-LINÉARITÉ DU SYSTÈME ET MÉTHODES D'IDENTIFICATIONS

### 2.1 Introduction

Dans ce chapitre, on discute du problème en général. On considère la non-linéarité de la branche de contrôle et ses conséquences. Ensuite, on définit le problème d'identification et on présente des techniques d'identifications linéaires et non-linéaires qui ont été couramment utilisées jusqu'à ce jour.

### 2.2 Non-linéarité de la branche de contrôle

Il est certain que la non-linéarité de la branche de contrôle, illustrée dans la figure 1.1, est confinée à l'arrangement actuateur piézo/poutre/capteur PVDF car les convertisseurs analogique/numérique et numérique/analogique, l'amplificateur haute tension et le préamplificateur, sont reconnus pour leur linéarité. La non-linéarité peut être due à plusieurs facteurs. Elle peut être due à une relation force-voltage non-linéaire ou à l'hystérésis dans l'actuateur piézo-électrique. Une élasticité mécanique non-linéaire de l'assemblage piézo/colle/poutre pourrait aussi être une source de non-linéarité. L'équation de Duffing, équation 2.1, est souvent utilisée pour représenter un oscillateur mécanique avec une élasticité non-linéaire :

$$m \ddot{\delta}(t) + D \dot{\delta}(t) + K \delta(t) + \gamma \delta^3(t) = f(t) \quad (2.1)$$

où  $m$  est la masse par unité de longueur,  $D$  est le coefficient d'amortissement,  $K$  est une constante d'élasticité linéaire,  $\gamma$  est une constante d'élasticité non-linéaire, et  $\delta(t)$  est le déplacement de la position repos pour une excitation  $f(t)$ . L'équation (2.1) de Duffing et sa solution théorique pour une excitation périodique sont relativement bien connues. Le lecteur intéressé peut consulter par exemple la référence [21, eq. 4.35].

Le capteur PVDF est connu pour être assez linéaire. Le circuit équivalent ainsi que la fonction de transfert, pour l'arrangement capteur PVDF/préamplificateur, sont donnés dans l'annexe 2. On voit que cet arrangement agit comme un différentiateur en basse fréquence.

### 2.3 Distorsion harmonique et fréquence d'échantillonnage

Un système LTI excité par un sinusoïde de fréquence  $f$  donnerait à la sortie un autre sinusoïde de même fréquence. Par contre, un système non-linéaire excité par un sinusoïde de fréquence  $f$  produirait une sortie composée des fréquences  $f$ ,  $2f$ ,  $3f$ , et ainsi de suite. De plus, si l'entrée était composée des fréquences  $f_1$  et  $f_2$ , la sortie contiendrait des termes de *distorsion harmonique* aux fréquences  $2f_1$ ,  $2f_2$ ,  $3f_1$ ,  $3f_2$ , ... et des termes de *distorsion intermodulation* aux fréquences  $f_1 \pm f_2$ ,  $2f_2 \pm f_1$ ,  $2f_1 \pm f_2$ , et ainsi de suite [18]. Ces propriétés sont importantes pour une évaluation globale de la non-linéarité.

La distorsion harmonique est reliée à la fréquence d'échantillonnage des signaux en temps continu. Un système non-linéaire excité par un signal de bande limitée  $B_i$  produirait une sortie de bande  $B_o > B_i$ . À première vue, on pourrait penser qu'il est essentiel d'échantillonner le signal de sortie du système à une fréquence au moins deux fois  $B_o$ . Pourtant, il a été démontré [20] qu'un système non-linéaire peut être identifié et compensé à la fréquence Nyquist du signal d'entrée. C'est-à-dire qu'on peut échantillonner au double de la fréquence la plus haute du signal d'excitation.

## 2.4 Identification des systèmes

L'identification des systèmes représente une partie très importante dans le contrôle des systèmes, surtout dans le contrôle actif. L'identification correspond souvent à la détermination d'une relation entrée-sortie d'un système.

Il y a plusieurs étapes dans l'identification [16] :

### 1. Choix du modèle

Un modèle est une description mathématique d'un système. Le modèle peut être :

- *paramétrique* ou *non-paramétrique*

modèle paramétrique : composé d'un nombre limité de quantités (caractéristiques).

modèle non-paramétrique : caractérisé par la mesure d'une fonction du système en un grand nombre de points.

- *linéaire* ou *non-linéaire*

- *boîte noire* ou *boîte blanche*

modèle de boîte blanche : basé sur une connaissance compréhensive du fonctionnement interne.

modèle de boîte noire : modèle comportemental basé sur des observations entrée-sortie.

### 2. Choix d'une structure

- Cela peut être, par exemple, le choix de l'ordre d'un filtre FIR ou le choix d'une structure parmi d'autres.

### 3. Estimation des paramètres

Des mesures sur le système physique sont utilisées pour déterminer les paramètres. Souvent on cherche à optimiser une fonction de performance, par exemple l'erreur de l'estimation, par rapport aux paramètres du système.

#### 4. Vérification du modèle

- Vérifier si les résultats prédits par le modèle correspondent assez bien aux observations sur le système physique.

### 2.5 Identification par filtrage adaptatif linéaire

Un système linéaire et temps-invariant (LTI) à mémoire peut être estimé par un filtre FIR. Les paramètres du filtre peuvent être estimés par les observations entrée-sortie du système. Figure 2.1 montre un modèle d'identification.

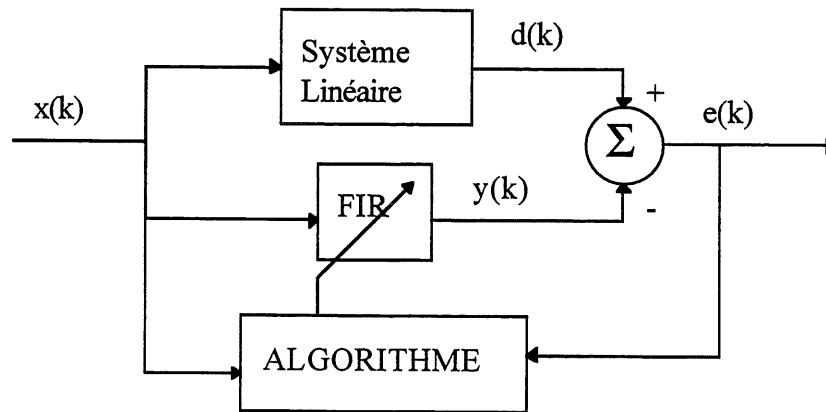


Figure 2.1 Identification par filtre transversal adaptatif

$x(k)$  est l'entrée et  $d(k)$  la sortie du système.  $y(k)$  est la sortie du filtre FIR qui estime le système LTI. Le but de l'identification est de trouver les paramètres optimaux du filtre FIR qui minimisent l'énergie de l'erreur  $e(k)$  de l'estimation.

Les paramètres optimaux,  $\mathbf{h}_{\text{opt}}$ , sont donnés par l'équation de Wiener-Hopf [5] :

$$\mathbf{h}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{p} \quad (2.2)$$



où  $\mathbf{R}$  est la matrice d'autocorrélation du signal  $x(k)$ :

$$\mathbf{R} = E\{\mathbf{x}_k \mathbf{x}_k^T\} \quad (2.3)$$

$$\text{où } \mathbf{x}_k = [x(k) \ x(k-1) \ x(k-2) \ \dots \ x(k-N+1)]^T \quad (2.4)$$

et  $\mathbf{p}$  est le vecteur d'intercorrélation entre le signal  $x(k)$  et le signal  $d(k)$ :

$$\mathbf{p} = E\{d(k) \mathbf{x}_k\} \quad (2.5)$$

Au lieu de chercher à calculer  $\mathbf{h}_{\text{opt}}$  par l'équation (2.2), la technique du filtrage adaptatif permet d'ajuster les paramètres du filtre FIR à chaque échantillon d'après un certain algorithme. Si l'algorithme a été bien conçu, les paramètres  $\mathbf{h}$  du filtre vont converger vers  $\mathbf{h}_{\text{opt}}$  après un certain nombre d'itérations.

Plusieurs algorithmes ont été développés et utilisés [5]. Les plus communs sont:

- 1) la méthode dite du gradient dont une forme est la méthode de LMS (*least mean squares*),
- 2) la méthode des moindres carrés récurrents (RLS pour *recursive least squares*).

La méthode du LMS a l'avantage de posséder une très faible charge de calcul. Cependant, elle n'est pas celle qui donne une plus rapide convergence. Par contre, l'algorithme RLS est beaucoup plus rapide mais demande une plus grande charge de calcul.

## 2.6 Identification par séries de Volterra

Les séries de Volterra ont été étudiées pour la première fois par Vito Volterra dans les années 1880. Pourtant, ce n'est qu'en 1958 qu'elles furent

utilisées par Norbert Wiener pour modéliser la relation entrée-sortie d'un système non-linéaire [5].

La relation entrée-sortie d'un système non-linéaire à mémoire, invariant dans le temps, et représentée en temps discret, peut être décrite par l'équation (2.6) [13] :

$$y(n) = h_0 + \sum_{k=1}^{\infty} \bar{h}_k[x(n)] \quad (2.6)$$

où  $x(n)$  est l'entrée et  $y(n)$  la sortie,  $h_0$  est une constante, et  $\bar{h}_k[x(n)]$  est donnée par:

$$\bar{h}_k[x(n)] = \sum_{i_1=0}^{N-1} \cdots \sum_{i_k=0}^{N-1} h_k(i_1, \dots, i_k) x(n-i_1) \dots x(n-i_k) \quad (2.7)$$

Les équations (2.6) et (2.7) constituent la série de Volterra. Chaque  $\bar{h}_k[x(n)]$  est appelé un noyau de Volterra.

Notons que pour  $k=1$ , le terme  $h_1(i_1)$  est la réponse impulsionnelle linéaire usuelle, et le terme  $h_k(i_1, \dots, i_k)$  peut être considéré comme une réponse impulsionnelle d'ordre  $k$ . Dans une série de Volterra d'ordre  $K$ , la limite supérieure dans l'équation (2.6) est remplacée par  $K$ . Pour  $K=2$ , on obtient une série de Volterra d'ordre 2. Les équations (2.6) et (2.7) se réduisent alors à :

$$\begin{aligned} y(n) = h_0 &+ \sum_{i_1=0}^{N-1} h_1(i_1) x(n-i_1) \\ &+ \sum_{i_1=0}^{N-1} \sum_{i_2=0}^{N-1} h_2(i_1, i_2) x(n-i_1) x(n-i_2) \end{aligned} \quad (2.8)$$

Un filtre adaptatif basé sur les séries de Volterra peut être décomposé de la manière suivante [5] :

1) un augmenteur d'état Volterra non-linéaire qui combine l'ensemble des entrées  $x(n)$ ,  $x(n-1)$ , ...,  $x(n-N+1)$  pour produire un plus grand ensemble d'états  $u_0$ ,  $u_1$ , ...,  $u_{q-1}$  où  $q > N$ . Par exemple, le vecteur augmenté pour un système quadratique avec  $N = 3$  a la forme  $\mathbf{u} = [1, x(n), x(n-1), x(n-2), x(n)^2, x(n)x(n-1), x(n)x(n-2), x(n-1)^2, x(n-1)x(n-2), x(n-2)^2]^T$

2) un filtre linéaire adaptatif qui opère sur le vecteur d'état  $\mathbf{u}$  pour produire une estimation  $y(n)$  d'une réponse désirée  $d(n)$ . L'algorithme LMS peut être utilisé pour faire l'adaptation. Ceci est illustré dans la figure 2.2.

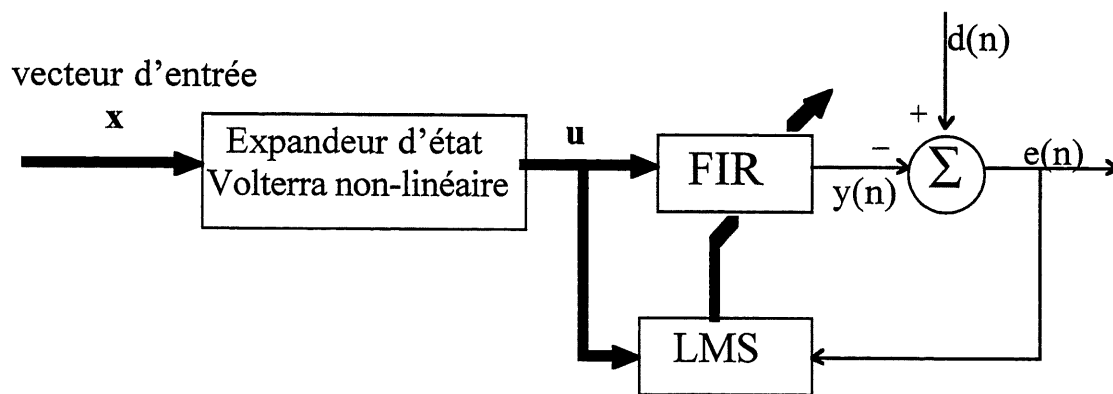


Figure 2.2 Filtre Volterra adaptatif

Plusieurs applications des filtres Volterra quadratiques ont été étudiées [13]. Voici quelques exemples : la réduction adaptative d'écho, la réduction adaptative d'interférence et l'égalisation adaptative des canaux de communications.

## 2.7 Réseaux de neurones de type feedforward

Depuis leur introduction dans les années 1950, beaucoup de recherches ont été faites sur les réseaux de neurones. Le résultat : ce champ d'étude est maintenant très développé, des outils analytiques très perfectionnés sont disponibles, et les applications sont nombreuses. Le but ici est de présenter sommairement ces puissants outils.

Les réseaux de neurones ont été très utilisés dans beaucoup de domaines. Les possibilités sont grandes. Dans [6], ces réseaux sont présentés comme des outils pour simultanément tenir compte du comportement non-linéaire, non-stationnaire et non-Gaussien d'un processus ou d'un système. Dans les exemples présentés, des utilisations sont faites pour la modélisation du chaos, la détection des signaux dans un environnement non-stationnaire et la compression d'image.

On distingue deux types de réseaux de neurones: les réseaux *feedforward* à multiples couches et les réseaux récurrents. Dans le premier type, il n'y a que des chemins directs entre les entrées et les sorties du réseau, c'est-à-dire que les sorties sont des fonctions non-linéaires des entrées seulement. Dans le deuxième type, il existe aussi des chemins de retour entre les sorties et les entrées du réseau, c'est-à-dire que les sorties du réseau sont des fonctions non-linéaires des entrées ainsi que des sorties précédentes. On se concentre sur les réseaux de types feedforward pour les raisons suivantes :

- À cause des chemins de retour entre les sorties et les entrées dans les réseaux récurrents, il n'est pas possible de garantir la stabilité du réseau durant l'apprentissage. Par contre, la stabilité d'un réseau de type feedforward est assurée.

- La plupart des algorithmes performants pour l'apprentissage des réseaux de neurones ont été développés particulièrement pour les réseaux de types feedforward.
- On peut utiliser des réseaux de types feedforward pour faire des identifications (même si les modèles sur lesquels on se base sont récurrents) en utilisant une structure série-parallèle (voir la section 2.8).

La figure 2.3 illustre un réseau de type feedforward de deux couches dont une est cachée.

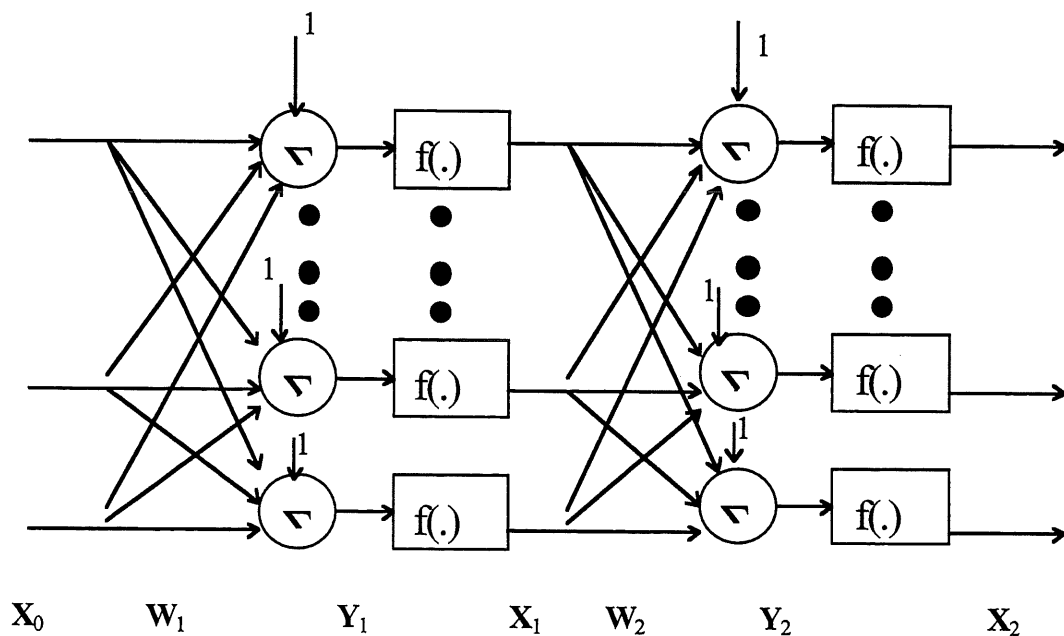


Figure 2.3 Réseau feedforward de deux couches

Le réseau accepte un vecteur d'entrée  $X_0$ , où :

$$X_0 = [x_{01}, x_{02}, \dots, x_{0N_0}]^T \quad (2.9)$$

La première couche (cachée) comprend  $N_1$  cellules décrites par :

$$\mathbf{Y}_1 = \mathbf{W}_1 \cdot [1, \mathbf{X}_0^T]^T \quad (2.10)$$

$$\mathbf{X}_1 = f[\mathbf{Y}_1] \quad (2.11)$$

où  $\mathbf{W}_1$  est la matrice des poids dans la première couche, et  $f$  est une fonction non-linéaire de type sigmoïde ou tangent hyperbolique.

La  $k^{\text{ième}}$  couche comprend  $N_k$  cellules décrites par:

$$\mathbf{Y}_k = \mathbf{W}_k \cdot [1, \mathbf{X}_{k-1}^T]^T \quad (2.12)$$

$$\mathbf{X}_k = f[\mathbf{Y}_k] \quad (2.13)$$

D'après [7], un réseau de neurones de type *feedforward* à multiples couches est capable d'estimer n'importe quelle fonction mesurable à n'importe quelle précision désirée, si on n'impose pas une limite supérieure à la taille du réseau. Dans un cas pratique, cette dernière est limitée, donc la précision de l'estimation est limitée aussi. Tout manque de succès dans l'application d'un tel réseau serait dû à un apprentissage inadéquat, un nombre insuffisant de couches cachées, ou à l'inexistence d'une relation déterministe entre l'entrée et la sortie désirée.

L' apprentissage consiste à présenter un ensemble de patrons d'entrée et un ensemble de sorties désirées correspondant, et à ajuster les paramètres du réseau afin d'optimiser une fonction de performance de l'approximation, telle que l'énergie de l'erreur entre la sortie désirée et la sortie du réseau. Ce processus est répété autant de fois qu'il faut pour converger vers la solution optimale. La méthode traditionnellement utilisée pour les réseaux feedforward est l'algorithme de rétropropagation standard. Bien que cet algorithme jouisse d'un poids de calcul assez faible, il possède plusieurs désavantages. Le premier est la très faible vitesse

d'apprentissage. Deuxièmement, le succès de l'algorithme repose sur des paramètres qui sont assez difficiles à régler. Dans beaucoup de cas, il faut essayer l'apprentissage à plusieurs reprises, avec diverses valeurs pour ces paramètres, pour avoir des bons résultats. Finalement, avec cet algorithme, le réseau a souvent tendance à aboutir dans des minimums locaux, ce qui donne des solutions assez loin de la solution optimale.

Un algorithme qui donne une convergence beaucoup plus rapide que l'algorithme de rétropropagation standard, et dont les paramètres d'apprentissage sont beaucoup plus facile à régler, fut présenté en 1992 [15]. Cet algorithme est décrit sommairement ci-dessous :

- 1) Initialiser les poids et les matrices  $\mathbf{R}^{-1}$  pour chaque couche.
- 2) Présenter un patron d'entrée et une sortie désirée.
- 3) Faire propager le patron dans le réseau.
- 4) Calculer les gains de Kalman et mettre à jour  $\mathbf{R}^{-1}$  pour chaque couche.
- 5) Calculer les signaux d'erreur de chaque couche par la rétropropagation.
- 6) Calculer les nouveaux poids pour chaque couche.
- 7) Si la convergence n'est pas satisfaisante retourner à étape 2.

## 2.8 Identifications par réseaux de neurones

Quatre modèles pour la représentation des systèmes SISO sont donnés dans [10].

$$\text{Modèle I: } y_p(k+1) = \sum_{i=0}^{n-1} \alpha_i y_p(k-i) + g[u(k), u(k-1), \dots, u(k-m+1)] \quad (2.14)$$

Modèle II:

$$y_p(k+1) = f[y_p(k), y_p(k-1), \dots, y_p(k-n+1)] + \sum_{i=0}^{m-1} \beta_i u(k-i) \quad (2.15)$$

Modèle III:

$$\begin{aligned} y_p(k+1) &= f[y_p(k), y_p(k-1), \dots, y_p(k-n+1)] \\ &+ g[u(k), u(k-1), \dots, u(k-m+1)] \end{aligned} \quad (2.16)$$

Modèle IV:

$$\begin{aligned} y_p(k+1) &= f[y_p(k), y_p(k-1), \dots, y_p(k-n+1); \\ &u(k), u(k-1), \dots, u(k-m+1)] \end{aligned} \quad (2.17)$$

où  $[u(k), y_p(k)]$  représente la paire entrée-sortie du système SISO au temps  $k$ ,  $y_p(k+1)$  représente la sortie au temps  $k+1$  dans le futur, et  $m \leq n$ .

Les quatre modèles sont représentés sous la forme de diagrammes dans les figures 2.4(a) à 2.4(d). On suppose que  $f(\cdot)$  et  $g(\cdot)$  sont des fonctions sur lesquelles on peut prendre les dérivées par rapport à chacun des arguments. Dans chaque modèle, la sortie au temps  $k+1$  dépend des valeurs des  $n$  sorties précédentes  $y_p(k-i)$  (ou  $i = 0, 1, \dots, n-1$ ) ainsi que des valeurs des  $m$  entrées  $u(k-j)$  (ou  $j = 0, 1, \dots, m-1$ ). Pour le modèle I, la sortie  $y_p(k+1)$  dépend d'une façon non-linéaire des entrées  $u(k-j)$ , mais la dépendance des sorties précédentes  $y_p(k-i)$  est



linéaire. Donc, ce modèle est approprié pour des systèmes avec une non-linéarité à l'entrée seulement. Pour le modèle II, la sortie  $y_p(k+1)$  dépend d'une façon linéaire des entrées  $u(k-j)$ , mais la dépendance sur les sorties précédentes  $y_p(k-i)$  est non-linéaire. Ce modèle est utile pour représenter des systèmes avec une non-linéarité à la sortie seulement, par exemple l'oscillateur de Duffing. Dans le cas du modèle III, la sortie  $y_p(k+1)$  dépend de façon non-linéaire des entrées, ainsi que des sorties précédentes. Ce modèle est une généralisation des deux premiers (I et II). Cependant, il est important de noter que la dépendance des entrées et des sorties précédentes est séparable. Le dernier modèle (IV) est le plus général et il englobe les trois autres. Malgré sa plus grande généralité, le modèle IV est le moins attrayant du point de vue analytique. Les trois premiers modèles sont plus pratiques en raison de leur simplicité.

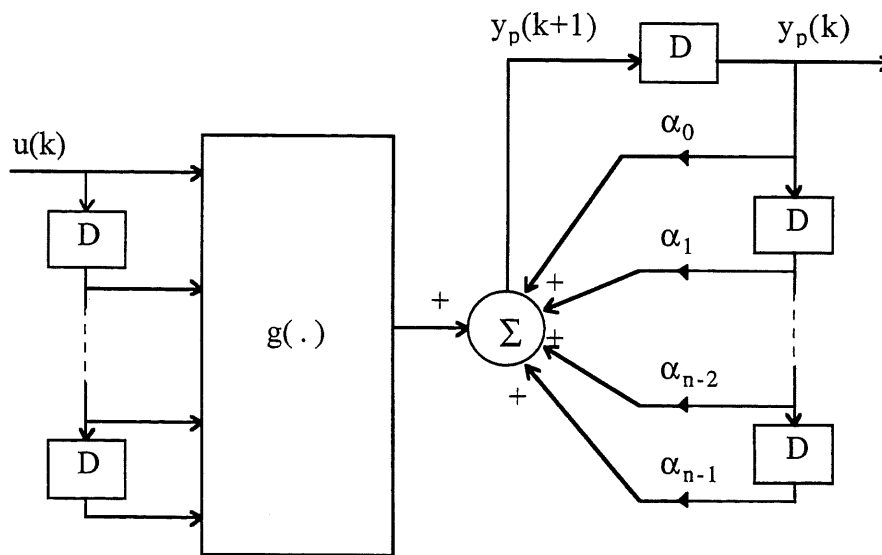


Figure 2.4(a) Modèle I

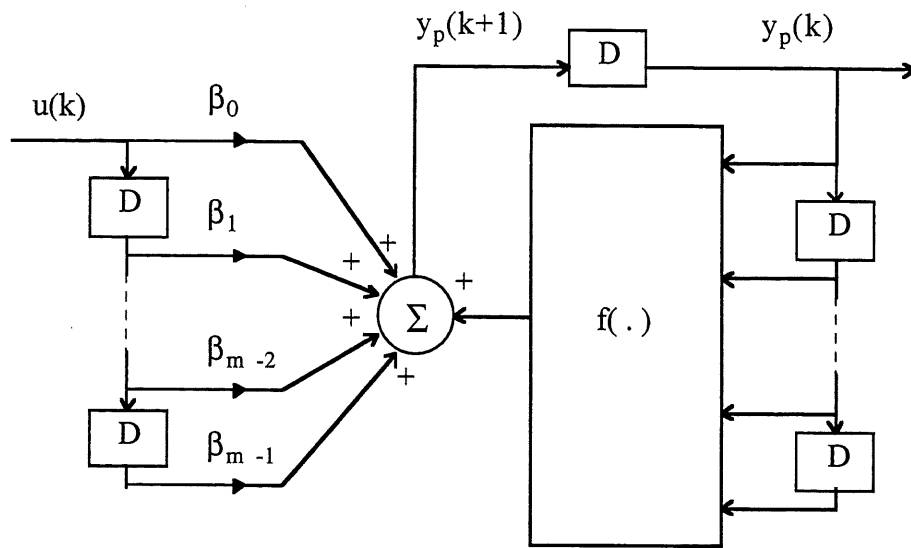


Figure 2.4(b) Modèle II

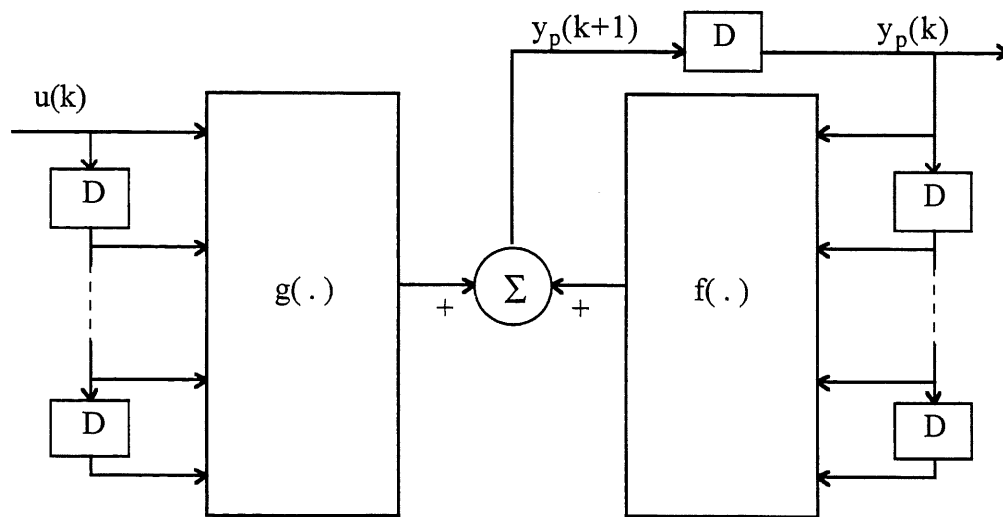


Figure 2.4(c) Modèle III

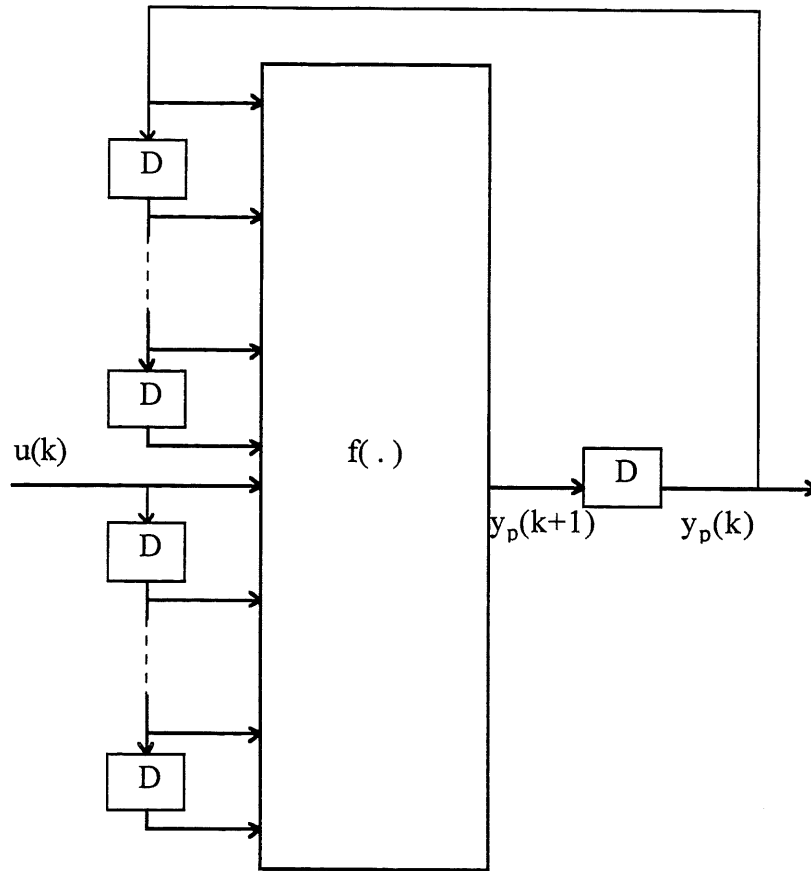


Figure 2.4(d) Modèle IV

Leur habilité à représenter des fonctions générales multivariables rendent les réseaux de neurones de type feedforward des candidats idéaux pour estimer les fonctions  $f(\cdot)$  et  $g(\cdot)$  dans les quatre modèles précédents [10]. Les réseaux feedforward peuvent donc être vus comme des composants, tout comme les retards purs, les multiplicateurs, et les additionneurs qui servent à construire des modèles plus complexes. Les modèles peuvent être eux-mêmes récurrents. Aussi, les réseaux de neurones récurrents peuvent être vus comme des modèles récurrents construits avec des réseaux feedforward et des éléments de retard [10].

Le problème d'identification consiste à choisir un modèle approprié, et à ajuster les paramètres de ce modèle, pour optimiser une fonction de performance basée sur l'erreur entre les sorties du système physique et du modèle. L'ajustement des paramètres est fait de manière itérative. Cependant, il faut prendre certaines mesures pour simplifier l'algorithme d'apprentissage et pour assurer la convergence vers la solution optimale.

Il y a deux modèles d'identification [10] : 1) modèle d'identification en parallèle, et 2) modèle d'identification série-parallèle. Un exemple d'une identification en parallèle est donné dans la figure 2.5.

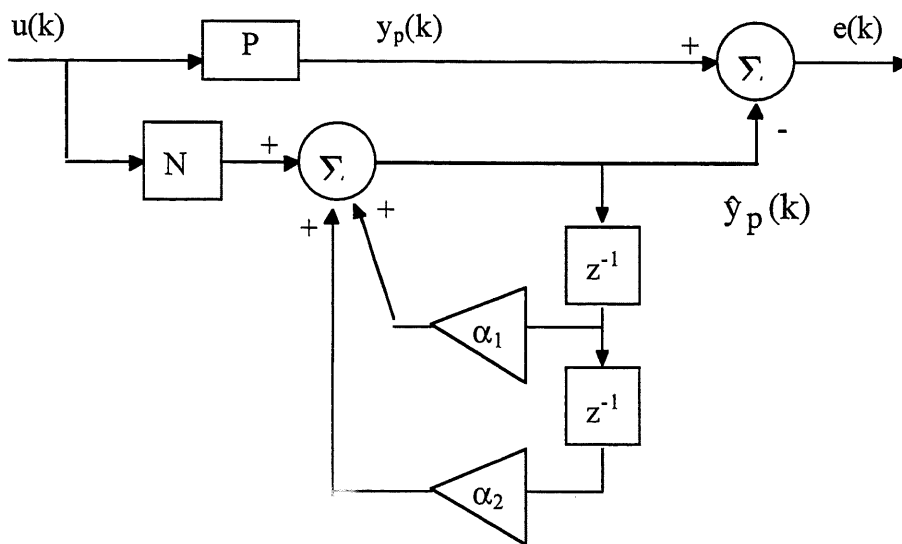


Figure 2.5 Identification en parallèle

Dans cet exemple, on essaye d'identifier un système non-linéaire, P, par le modèle suivant :

$$\hat{y}_p(k) = \alpha_1 \cdot \hat{y}_p(k-1) + \alpha_2 \cdot \hat{y}_p(k-2) + N[u(k)] \quad (2.18)$$

L'identification consiste à trouver les paramètres du réseau  $N$ , ainsi que  $\alpha_1$  et  $\alpha_2$ . On utilise ici la rétropropagation basée sur l'erreur  $e(k)$  entre la sortie du système  $P$  et la sortie du modèle.

Dans une identification série-parallèle du même système  $P$ , la sortie du système est retournée dans le modèle d'identification, comme démontrée dans la figure 2.6. Cela implique que le modèle d'identification, dans ce cas-là, est la suivante :

$$\hat{y}_p(k) = \alpha_1 \cdot y_p(k-1) + \alpha_2 \cdot y_p(k-2) + N[u(k)] \quad (2.19)$$

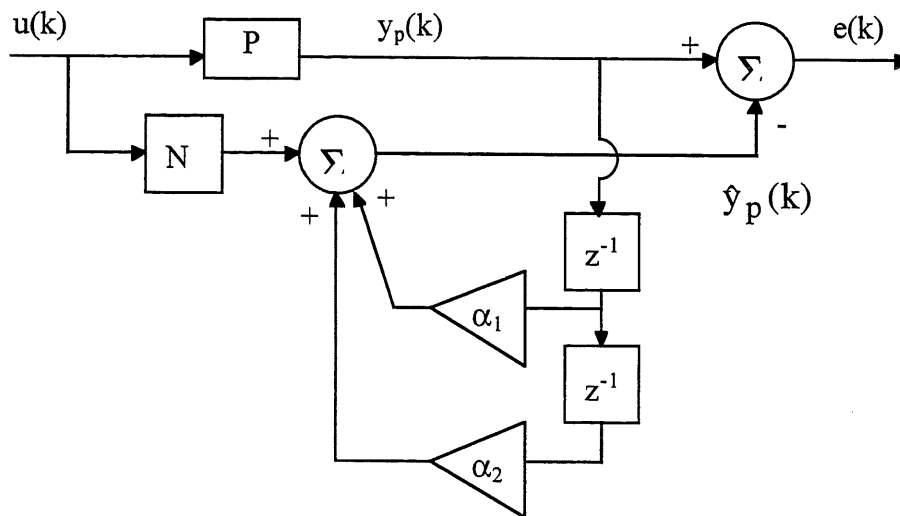


Figure 2.6 Identification série-parallèle

D'après [10], le modèle d'identification série-parallèle possède plusieurs avantages sur l'identification en parallèle. Le premier avantage est relié à la stabilité. Dans le cas d'une identification en parallèle, il n'y a aucune garantie que le modèle sera stable durant l'adaptation. C'est même très probable que le modèle devienne instable, ce qui rend l'apprentissage très difficile. Par contre, avec une

identification série-parallèle, la stabilité durant l'apprentissage est assurée. Aussi, en utilisant un modèle série-parallèle, l'algorithme d'apprentissage est beaucoup plus simple, puisqu'on a deux signaux de référence ( $x(k)$  et  $y_p(k)$ ) et les problèmes de minimums locaux sont moins graves. Par contre, l'algorithme d'apprentissage pour un modèle en parallèle peut être extrêmement complexe.

Cependant, le modèle d'identification série-parallèle a aussi son inconvénient. Dans tous les problèmes d'identification, la sortie  $y_p(k)$  du système physique est contaminée par un bruit qui est souvent aléatoire et non-corrélé à l'entrée. En retournant la sortie bruitée du système physique ( $y_p(k) + \text{bruit}$ ) dans le modèle, lors d'une identification série-parallèle, on altère quelque peu le critère de l'optimisation. En effet, une identification par un modèle série-parallèle cherche non seulement à trouver la relation entre l'entrée  $x(k)$  et  $y_p(k)$ , mais elle essaye aussi à prédire le bruit à la sortie du système. Le résultat est que l'ensemble de paramètres du modèle converge vers une solution biaisée. Plus le bruit est important, plus grand sera le biais.

En considérant les avantages et les désavantages que nous offrent les deux modèles d'identification, on préfère le modèle série-parallèle, surtout pour sa stabilité et son apprentissage beaucoup moins complexe.

## 2.9 Bruit dans le système physique

Le bruit existe dans la plupart des systèmes physiques. Pour le système étudié, le bruit est présent dans les sous-systèmes électroniques et il provient aussi de l'environnement (vibrations mécaniques). La présence du bruit limite la précision des identifications du système.

Dans un test, l'actuateur piézo-céramique est laissé en circuit ouvert et la sortie du convertisseur A/N est observée. Un bloc de 2048 échantillons sur le signal de sortie est montré dans la figure 2.7(a), et son spectre d'énergie est donné dans la figure 2.7(b).

On remarque que le bruit est assez important. Pour une comparaison, l'amplitude moyenne du signal de sortie est d'environ 6000 lors des identifications. Aussi, l'énergie du bruit est concentrée en bande étroite autour de 15 Hz. Il est évident que la présence du bruit limiterait beaucoup la performance des identifications. Il est donc très important de vérifier la validité des modèles après les identifications.

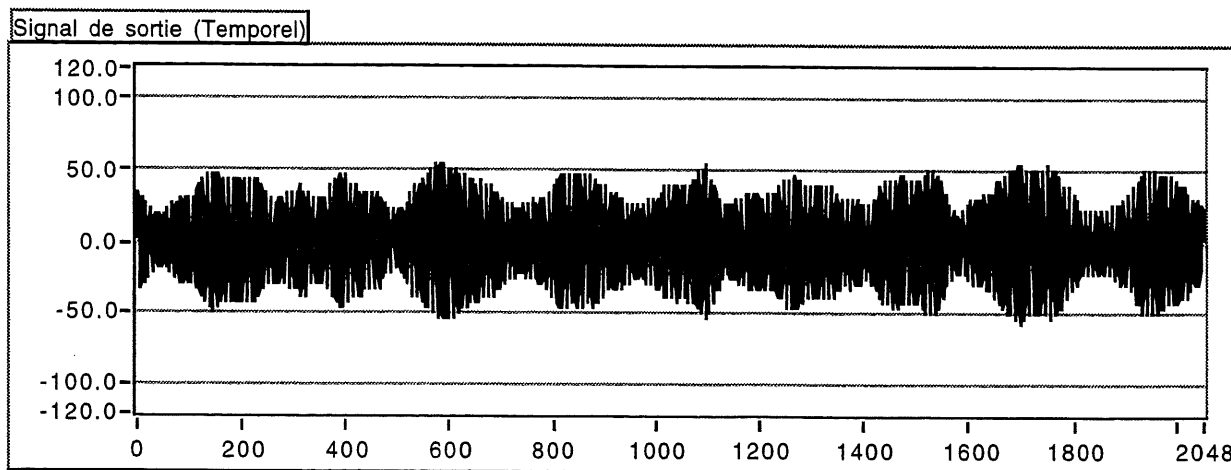


Figure 2.7(a) Bruit à la sortie avec l'actuateur en circuit ouvert

Note: l'échelle en y est doit être multipliée par  $91\mu\text{V/unité}$  pour obtenir une échelle en Volts

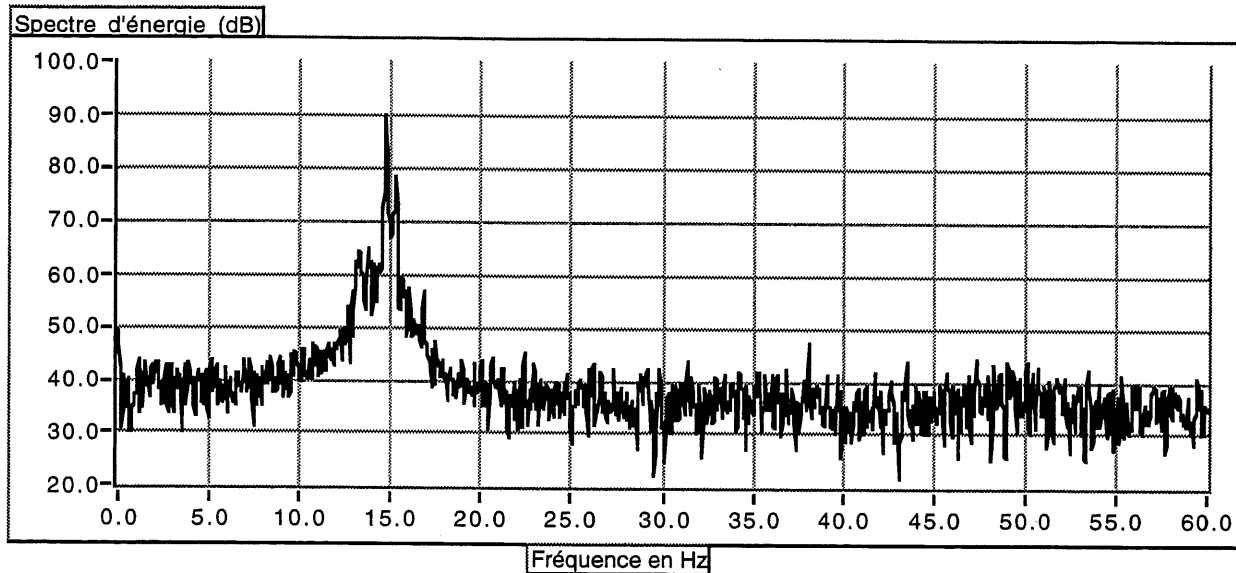


Figure 2.7(b) Spectre d'énergie du bruit

## 2.10 Conclusion

On retient certains points importants présentés dans ce chapitre. Le reste de ce document repose beaucoup sur ces observations.

Premièrement, la réponse du système pour des excitations en harmonique peut être utilisée pour faire une évaluation globale de la non-linéarité.

Deuxièmement, l'identification d'un système non-linéaire, par un modèle numérique en temps discret, peut se faire à la fréquence d'échantillonnage égale à la fréquence Nyquist du signal d'excitation.



Troisièmement, des modèles basés soit sur les séries de Volterra, ou sur les réseaux de neurones, peuvent être utilisés pour une identification non-linéaire du système. On utilise ces derniers à cause de leur formulation plus générale.

Finalement, un modèle d'identification série-parallèle est préféré à un modèle d'identification en parallèle, car ce premier possède plusieurs avantages (stabilité garantie durant l'apprentissage, algorithme d'apprentissage plus simple, meilleure chance de converger vers une solution). L'inconvénient majeur avec cette méthode, c'est que le modèle identifié risque d'être biaisé due à la présence du bruit dans le signal de sortie.

### 3. ÉVALUATION DE LA NON-LINÉARITÉ GLOBALE PAR EXCITATION EN RÉGIME HARMONIQUE

#### 3.1 Introduction

C'est un fait fondamental que lorsqu'un système linéaire est excité par un sinus pur, sa réponse en régime permanent est aussi un sinus pur de la même fréquence. Cependant, ceci n'est pas le cas pour un système non-linéaire. Lorsqu'un système non-linéaire est excité par un sinus pur, sa réponse en régime permanent n'est pas en général un sinus pur. Dépendant de la nature de la non-linéarité, on peut obtenir des harmoniques à des fréquences multiples de la fréquence d'excitation, et on peut aussi observer la bifurcation de fréquence (le doublement de la période fondamentale) dans le cas d'un système chaotique.

Dans ce chapitre, on utilise ces faits pour démontrer l'existence d'une non-linéarité dans le système actuateur piézo/poutre/capteur PVDF, et pour faire une évaluation globale du taux de non-linéarité présente dans ce système.

#### 3.2 Procédure de l'expérimentation

L'arrangement expérimental pour faire une estimation de la non-linéarité globale, présente dans la branche de contrôle, est illustré dans la figure 3.1. Notons que P représente la branche de contrôle. P comprend l'amplificateur de haute tension, l'arrangement actuateur piézo-électrique/poutre/capteur PVDF, le préamplificateur et le convertisseur analogique-numérique. Le signal d'excitation,  $x(t)$ , est donné par:

$$x(t) = A \sin(2\pi f t) \quad (3.1)$$

où  $A$  est l'amplitude d'excitation arbitrairement choisie à 1.0 V et la fréquence d'excitation,  $f$ , est variée de 3 à 20 Hz.

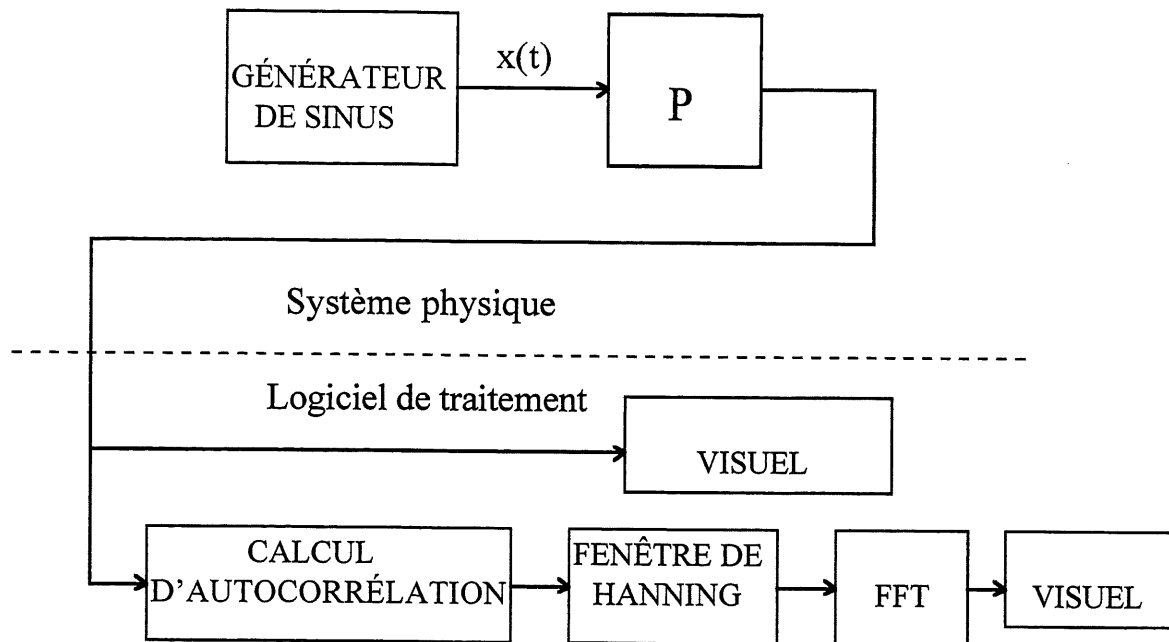


Figure 3.1 Arrangement pour évaluer la distorsion harmonique

Pour chaque fréquence d'excitation, on attend assez longtemps après le début du signal d'excitation pour que la réponse se stabilise, et on prend un bloc de 1024 échantillons du signal de sortie. On fait un calcul d'autocorrélation pour ce bloc de signal, on applique une fenêtre de Hanning sur la fonction d'autocorrélation pour réduire les effets de repliement, et ensuite on prend la transformée de Fourier du résultat pour finalement obtenir le spectre d'énergie. La partie amplitude du spectre est convertie en dB et ensuite affichée. Le signal de sortie est aussi affiché.

Notons qu'on n'utilise pas le convertisseur numérique-analogique intégré sur la carte DSKX, car la mémoire disponible sur la carte nous permet de n'envoyer qu'un signal de 4096 échantillons seulement comme excitation. Cela n'est pas suffisant si on veut observer la réponse en régime permanent. A la place,

on utilise une carte d'acquisition capable de générer continuellement un signal périodique.

### 3.3 Résultats

La partie amplitude des transformées de Fourier nous montre clairement l'effet de la non-linéarité du système. Puisque l'excitation est un sinus, la sortie du système est périodique. Donc, le spectre du signal de sortie est constitué des harmoniques à des fréquences discrètes. On a donc des pics dans le spectre représentant chacun un composant spectral: le fondamental et les autres harmoniques.

Les figures 3.2(a) à 3.2(d) nous montrent les spectres du signal de sortie pour les fréquences d'excitation de 3.0, 5.0, 7.5 et 17.0 Hz.

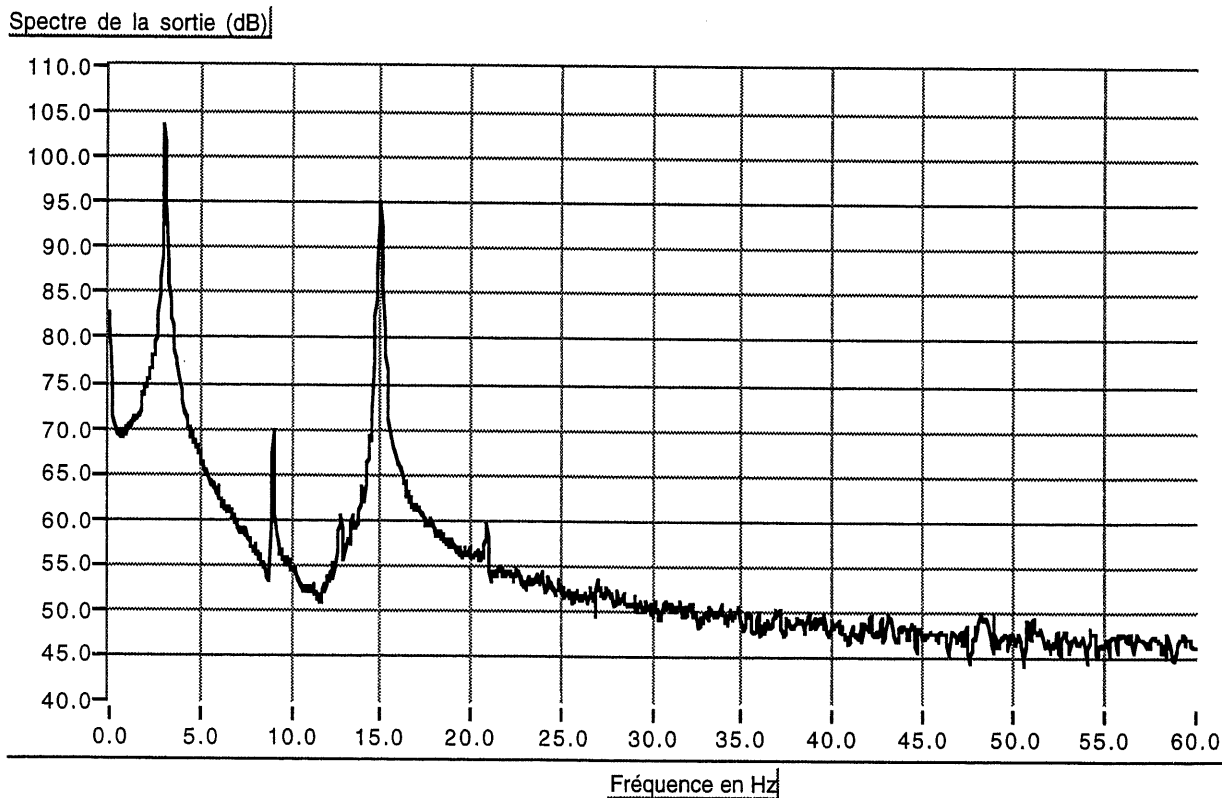


Figure 3.2(a) Spectre du signal de sortie pour une excitation de 3.0 Hz

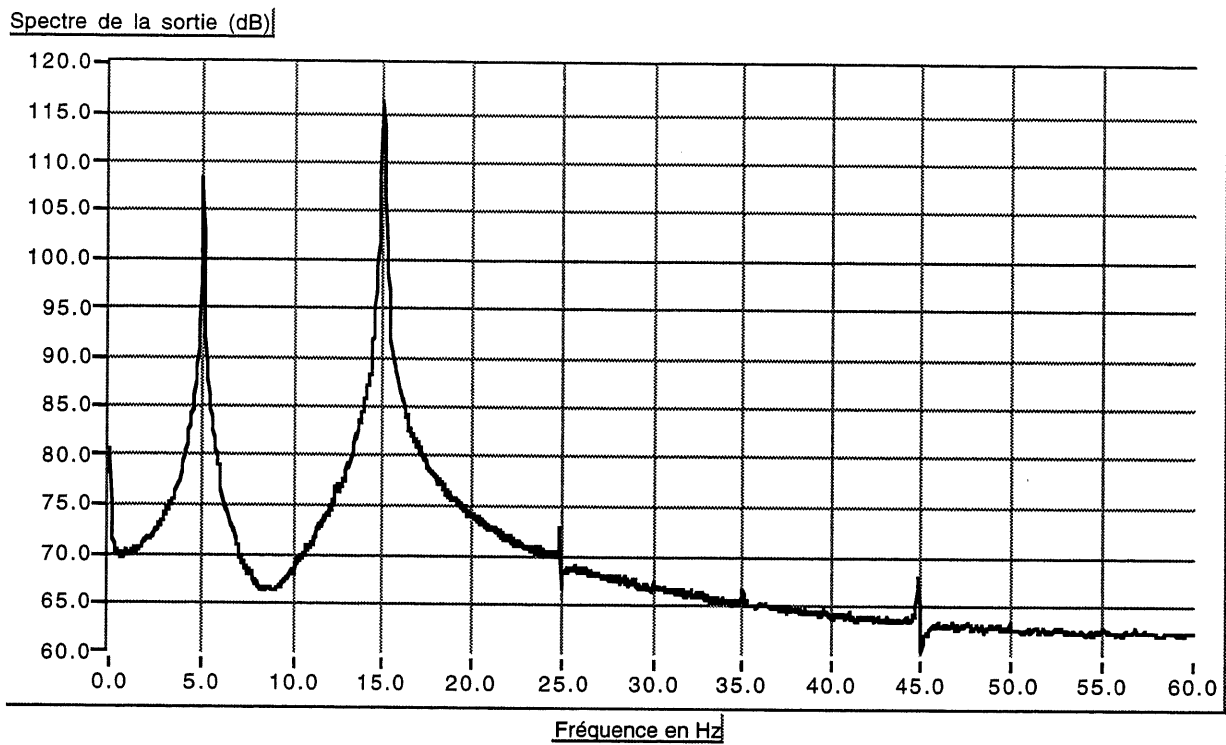


Figure 3.2(b) Spectre du signal de sortie pour une excitation de 5.0 Hz

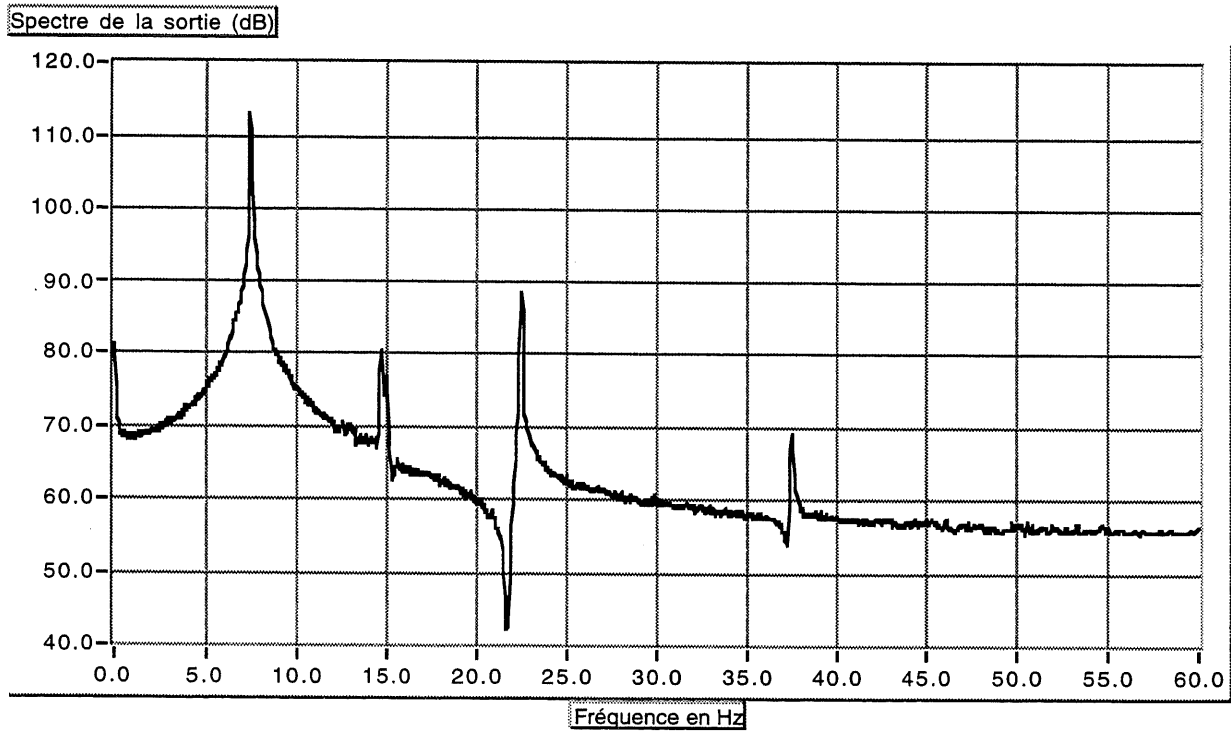


Figure 3.2(c) Spectre du signal de sortie pour une excitation de 7.5 Hz

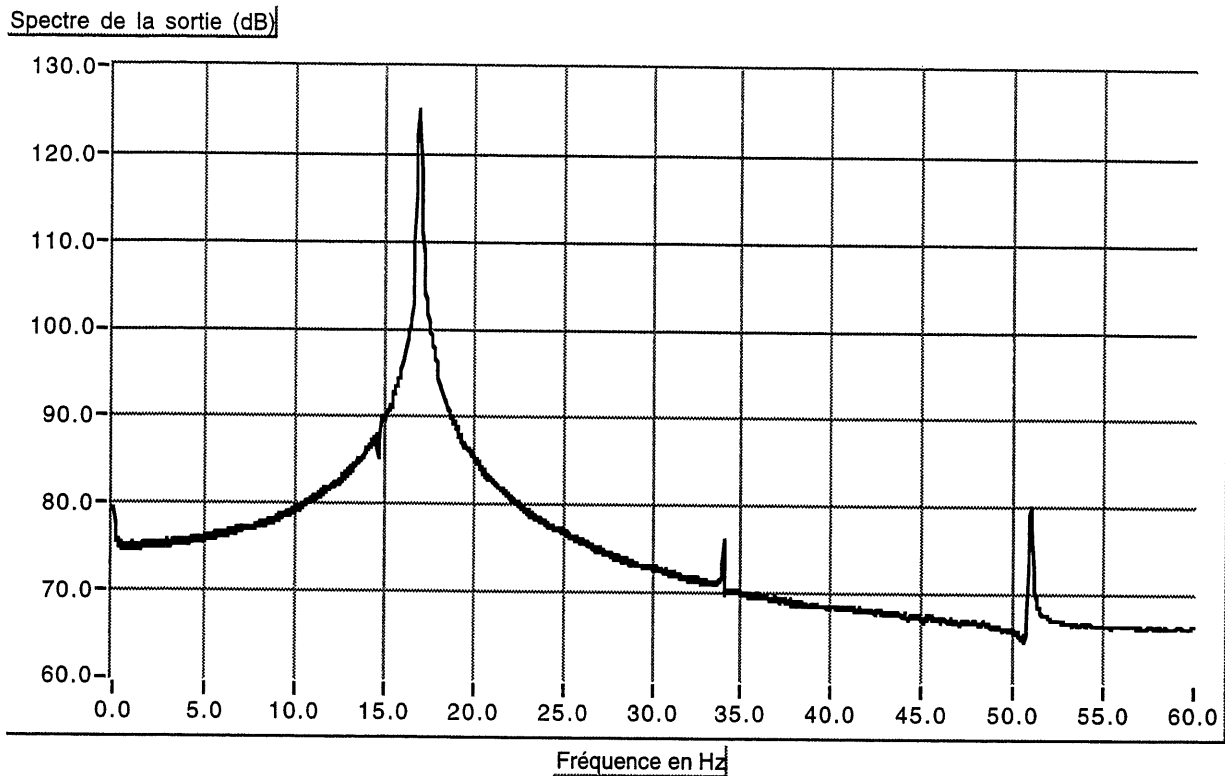


Figure 3.2(d) Spectre du signal de sortie pour une excitation de 17.0 Hz

Le tableau 3.1 nous montre en détail l'amplitude (en dB) de chaque composant, dans le spectre du signal de sortie, pour les fréquences de 3.0 Hz à 18.0 Hz. On évite les fréquences dans la voisinage de 15.0 Hz. En effet, la poutre est très résonante à cette fréquence, et une excitation trop proche de la fréquence résonante produit des oscillations de grandes amplitudes. Vue la façon dont la poutre est encastrée (voir la figure 3.3), la déflexion de la poutre est limitée par les bords  $P_1$  et  $P_2$ . Pour les fréquences d'excitation trop proches à 15 Hz, cette limite sur la déflexion forme une autre non-linéarité qui peut masquer celle de l'actuateur. De même, les fréquences d'excitations au-delà de 18.0 Hz ne sont pas très intéressantes puisqu'on échantillonne à 120.0 Hz, et la fréquence de coupure du filtre antirepliement est  $\approx 50$  Hz. Donc, les troisièmes harmoniques ne peuvent pas être observés pour les fréquences d'excitation au-delà de 18 Hz.

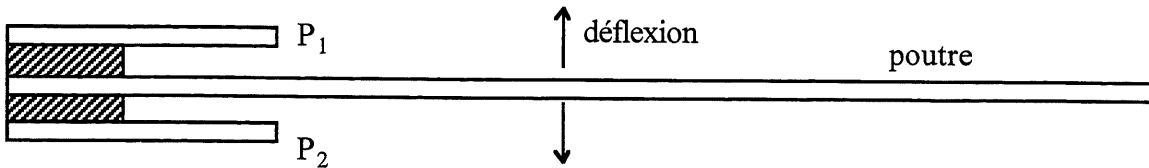


Figure 3.3 Vue de haut de la poutre

TABLEAU 3.1 AMPLITUDES DES HARMONIQUES DANS LE SIGNAL DE SORTIE POUR EXCITATION EN SINUS DE DIVERSES FRÉQUENCES

FRÉQUENCE ENTRÉE	PIC		2 <sup>ème</sup>		3 <sup>ème</sup>		5 <sup>ème</sup>	
	FONDAMENTAL		HARMONIQUE		HARMONIQUE		HARMONIQUE	
Hz	Hz	dB	Hz	dB	Hz	dB	Hz	dB
3.0	3.0	103	—	—	9.0	69	15.0	92
4.0	4.0	109	—	—	12.0	83	20.0	70
5.0	5.0	108	—	—	15.0	116	25.0	73
6.0	6.0	110	—	—	18.0	90	30.0	70
7.0	7.0	111	—	—	21.0	89	35.0	70
7.5	7.5	113	15.0	81	22.5	88	37.5	69
8.0	8.0	115	—	—	24.0	88	40.0	69
9.0	9.0	115	—	—	27.0	85	45.0	69
10.0	10.0	119	—	—	30.0	88	50.0	69
11.0	11.0	120	—	—	33.0	88	—	—
12.0	12.0	121	—	—	36.0	85	—	—
13.0	13.0	126	—	—	39.0	88	—	—
17.0	17.0	125	34.0	76	51.0	80	—	—
18.0	18.0	120	36.0	70	54.0	69	—	—

Il nous est difficile de faire une comparaison du taux de non-linéarité par cette table, car le système a une fonction de transfert très colorée, et selon sa fréquence, chaque harmonique est rehaussée ou atténuée. Il nous faut donc faire une compensation de la fonction de transfert. Cette fonction de transfert est obtenue en faisant une identification linéaire sur le système (voir chapitre suivant). Elle est montrée dans la figure 3.4. Aussi, il est préférable de normaliser les amplitudes de sorte que le composant fondamental soit ramené à 0 dB. Ceci nous permet de voir plus facilement le taux de distorsion. Le tableau 3.2 nous montre le niveau de chaque harmonique, après la compensation de la fonction de transfert, et la normalisation.

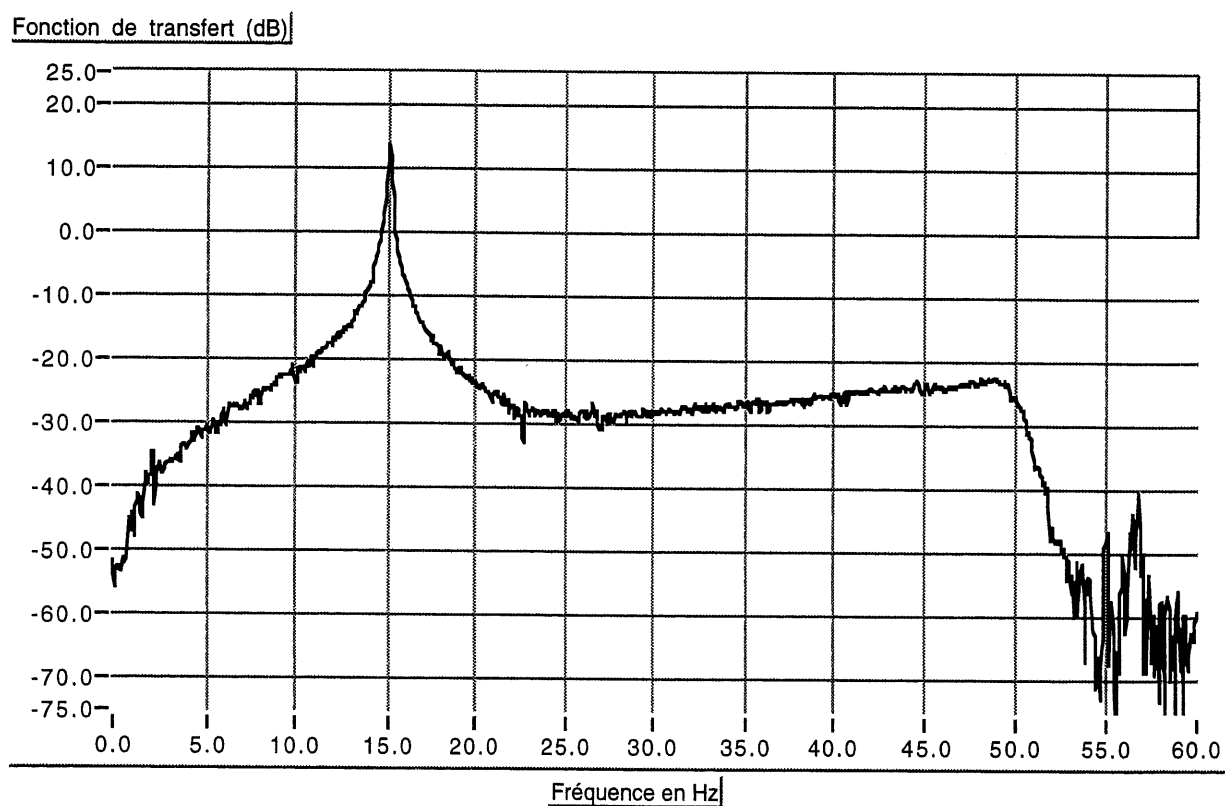


Figure 3.4 Fonction de transfert du système



TABLEAU 3.2 NIVEAUX DES HARMONIQUES APRÈS LA COMPENSATION DE  
L'EFFET DE LA FONCTION DE TRANSFERT ET NORMALISATION

FRÉQUENCE ENTRÉE	PIC		2 <sup>ème</sup>		3 <sup>ème</sup>		5 <sup>ème</sup>	
	FONDAMENTAL		HARMONIQUE		HARMONIQUE		HARMONIQUE	
Hz	Hz	dB	Hz	dB	Hz	dB	Hz	dB
3.0	3.0	0	—	—	9.0	-47	15.0	-58
4.0	4.0	0	—	—	12.0	-42	20.0	-48
5.0	5.0	0	—	—	15.0	-34	25.0	-37
6.0	6.0	0	—	—	18.0	-30	30.0	-41
7.0	7.0	0	—	—	21.0	-23	35.0	-41
7.5	7.5	0	15.0	-69	22.5	-21	37.5	-44
8.0	8.0	0	—	—	24.0	-23	40.0	-46
9.0	9.0	0	—	—	27.0	-23	45.0	-45
10.0	10.0	0	—	—	30.0	-25	50.0	-47
11.0	11.0	0	—	—	33.0	-25	—	—
12.0	12.0	0	—	—	36.0	-27	—	—
13.0	13.0	0	—	—	39.0	-28	—	—
17.0	17.0	0	34.0	-37	51.0	-25	—	—
18.0	18.0	0	36.0	-43	54.0	-10**	—	—

NOTE\*\*: Cette valeur est erronée due à une surcompensation de la fonction de transfert. En effet, la fonction de transfert, obtenue par une identification linéaire du système, tient aussi compte de la coupure de la fonction de transfert du convertisseur numérique/analogique sur la carte DSKX. Cependant, le pic à la fréquence 54.0 Hz représente un harmonique qui provient de la non-linéarité du système. Puisque cette non-linéarité est située dans la partie mécanique du

système, l'harmonique à la fréquence 54.0 Hz n'est pas influencé par la fonction de transfert en amont de la partie mécanique. Le gain du convertisseur N/A à la fréquence 54.0 Hz est de -22 dB (voir la figure 4.7). Donc, il y a eu une surcompensation de 22 dB. En faisant cette correction, on obtient une valeur de -32 dB pour le pic à 54.0 Hz.

### **3.4 Conclusion**

D'après les résultats obtenus par ces expérimentations, on voit que les excitations en régime harmonique ont été très utiles pour démontrer la distorsion harmonique, confirmant ainsi l'existence d'une importante non-linéarité dans le système.

On observe que la plupart des cas, les harmoniques sont impairs seulement (fondamental, 3<sup>ème</sup>, 5<sup>ème</sup>, etc.), et lorsque les harmoniques pairs sont présents, leur contribution est très faible. La bifurcation de la fréquence fondamentale (ou doublement de la période fondamentale) n'a pas été observée pour aucune fréquence d'excitation.

On remarque aussi que le niveau de distorsion varie entre environ -20 et -50 dB, dépendant de la fréquence d'excitation.

## 4. IDENTIFICATIONS DU SYSTÈME PAR MODÈLES LINÉAIRES

### 4.1 Introduction

Dans ce chapitre, on considère l'identification du système par des modèles linéaires. On commence par décrire sommairement l'algorithme LMS. Ensuite, on considère une identification du système par un modèle MA. On procède aussi à une identification des filtres dans les convertisseurs numérique-analogique et analogique-numérique par un modèle MA. Finalement, on considère une identification du système par un modèle ARMA. Ces identifications nous permettent d'évaluer la performance des identifications linéaires, pour qu'on puisse faire des comparaisons ultérieurement. La fréquence d'échantillonnage est à 120.0 Hz dans toutes les identifications.

### 4.2 Algorithme LMS pour filtre linéaire

Considérons le problème suivant. Le système  $P$  est excité par un signal de référence,  $x(n)$ , et sa réponse est le signal  $d(n)$ . En supposant que le système peut être modélisé par un filtre transversal,  $F$ , le problème d'identification est de trouver les paramètres,  $[h_0, h_1, h_2, \dots, h_{N-1}]$ , de  $F$  qui minimisent l'énergie du signal d'erreur d'identification,  $e(n)$ . Le problème est illustré dans la figure 4.1.

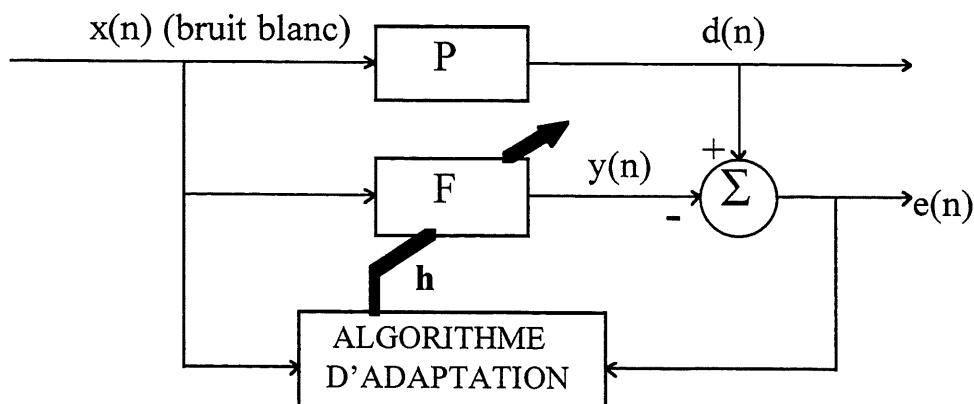


Figure 4.1 Identification par filtre linéaire transversal

Avec les définitions suivantes:

$$\mathbf{x}(n) = [x(n), x(n-1), x(n-2), \dots, x(n-N+1)]^T \quad (4.1)$$

$$\mathbf{h} = [h_0, h_1, h_2, \dots, h_{N-1}]^T, \quad (4.2)$$

la sortie du filtre F,  $y(n)$ , est donnée par:

$$y(n) = \mathbf{h}^T \mathbf{x}(n) \quad (4.3)$$

et le signal d'erreur par:

$$e(n) = d(n) - y(n). \quad (4.4)$$

Pour trouver le vecteur  $\mathbf{h}$  qui minimise l'énergie du signal  $e(n)$ , l'algorithme LMS est souvent utilisé. C'est un algorithme de filtrage adaptatif qui fonctionne sur chaque échantillon des signaux connus. L'algorithme se résume de la façon suivante.

- 1) Les poids dans le vecteur  $\mathbf{h}$  sont initialisés à zéro.
- 2) Pour chaque échantillon du signal  $x(n)$ , on calcule  $y(n)$  et  $e(n)$  par les équations 4.3 et 4.4.
- 3) Le vecteur  $\mathbf{h}$  est mis à jour à chaque itération d'après l'équation 4.5.
$$\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e(n) \mathbf{x}(n) \quad (4.5)$$
où  $\mu$  est le pas d'adaptation.
- 4) Si l'énergie de l'erreur  $e(n)$  est assez petit, on arrête l'adaptation de  $\mathbf{h}$ . Sinon on retourne à l'étape 2.

Si le modèle est représentatif, et que le pas d'adaptation est bien choisi, le vecteur  $\mathbf{h}$  converge vers une solution optimale  $\mathbf{h}_{opt}$ . La vitesse de convergence

dépend du pas d'adaptation. Si ce dernier est trop petit, la vitesse de convergence est très lente : il faut beaucoup d'itérations pour arriver à  $\mathbf{h}_{opt}$ . Par contre, si  $\mu$  est trop grand, l'algorithme est instable, et le vecteur  $\mathbf{h}$  diverge de  $\mathbf{h}_{opt}$ .

Dans une forme normalisée de l'algorithme, le pas d'adaptation est choisi comme [5]:

$$\mu(n) = \frac{\mu_{norm}}{\sum_{k=0}^{N-1} E\{x(n-k)^2\}} \quad (4.6)$$

où  $\mu_{norm} \leq 1.0$ . Bien qu'il existe d'autres algorithmes plus performants que le LMS, ce dernier possède une très faible complexité. Pour toutes les identifications par modèle linéaire faites dans ce travail, c'est cet algorithme qui est donc utilisé.

Il est important de bien définir un critère pour mesurer la performance de l'identification. C'est-à-dire qu'on doit définir un paramètre qui nous indique si on se rapproche ou si on s'éloigne de la solution optimale, et aussi, qui sert à évaluer le modèle, une fois que ce dernier a convergé. Le critère souvent utilisé est le rapport de l'énergie du signal d'erreur sur l'énergie du signal désiré. Le terme « convergence » est utilisé pour ce rapport d'énergie dans le reste de ce document. L'énergie du signal d'erreur doit être calculée sur une fenêtre de petite taille pour pouvoir suivre assez rapidement l'évolution de l'adaptation. Cependant, l'énergie du signal de sortie varie à cause de la résonance du système et doit être calculée sur une fenêtre assez large. Une façon de faire est de calculer l'énergie sur une fenêtre exponentielle. C'est-à-dire que l'énergie  $E(n)$  d'un signal  $s(n)$  quelconque au temps  $n$  est calculée de la façon suivante :

$$E(n) = a \cdot [s(n)]^2 + (1 - a) \cdot E(n - 1) \quad (4.7)$$

où  $E(n-1)$  est l'énergie au temps précédent et  $0 < a \ll 1.0$ . Une valeur  $a = 0.01$  est utilisée pour l'identification par modèle MA, et  $a = 0.005$  dans toutes les autres identifications.

### 4.3 Identification du système par modèle MA adaptatif

Une identification par un filtre FIR de 1024 coefficients, basée sur l'algorithme LMS normalisé, donne une convergence de  $-23 \pm 8$  dB. Plusieurs facteurs contribuent à cette importante erreur d'identification. D'abord, le système est très résonant et même 1024 coefficients ne suffisent pas à le modéliser. Ensuite, la non-linéarité du système et le bruit dans l'environnement forment d'autres contributions à l'erreur d'identification. La variation de 8 dB dans la convergence provient du fait que l'énergie du signal de sortie varie considérablement, à cause de la résonance du système (signal de sortie en bande étroite).

La réponse impulsionnelle du système, qui correspond aux coefficients du filtre FIR ordonnés, est montrée dans la figure 4.2. La fonction de transfert calculée avec ces coefficients est donnée dans la figure 4.3.

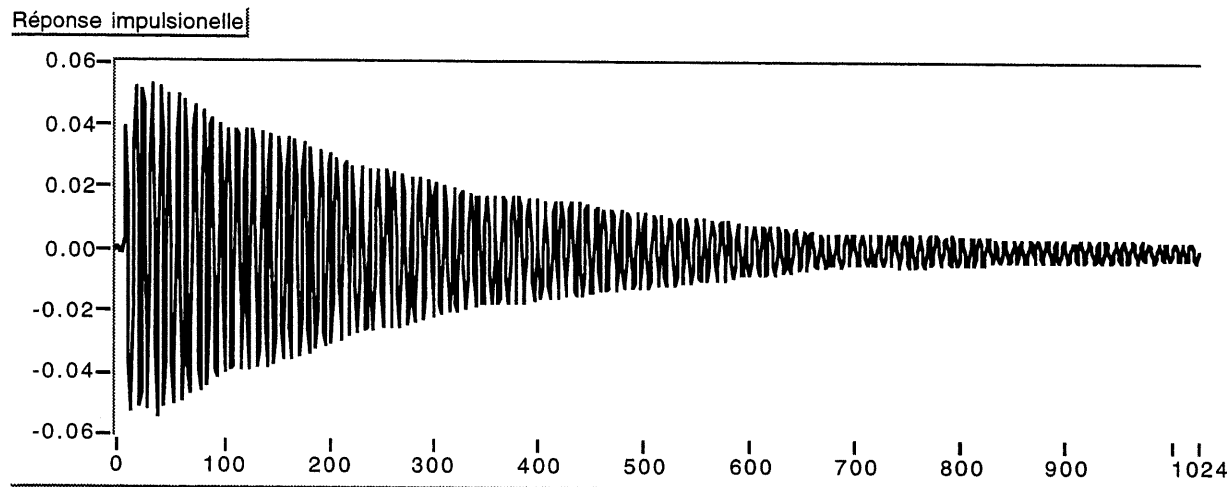


Figure 4.2 Réponse impulsionnelle identifiée

Fonction de transfert (dB)

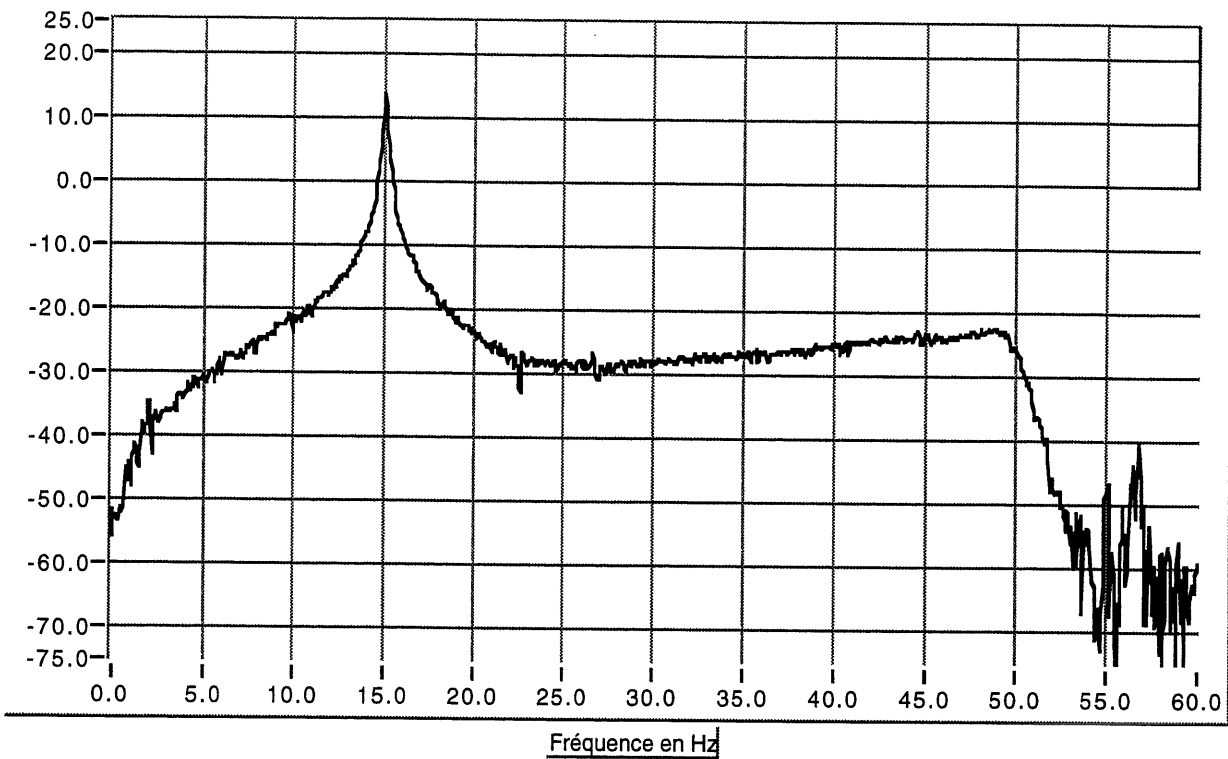


Figure 4.3 Fonction de transfert identifiée

Bien qu'une identification par un modèle FIR soit inadéquate, elle nous permet d'avoir une idée générale sur le comportement du système. Premièrement, en examinant la réponse impulsionnelle, c'est évident qu'un modèle ARMA (autoregressif avec moyenne ajustée) serait plus pertinent. Deuxièmement, en examinant la fonction de transfert du modèle, on s'aperçoit qu'il n'y a qu'un pôle très résonant à 15 Hz dans la bande 0 à 60 Hz. Donc, deux coefficients devraient suffire pour la partie autoregressif du modèle ARMA.

#### 4.4 Identification des filtres d'acquisitions par modèle MA

Dans cette section, on considère une identification FIR sur le sous-système composé des filtres pour les conversions numérique-analogique et analogique-numérique en série. Le but est de vérifier que ces filtres peuvent être représentés par des FIR, bien qu'ils soient elliptiques, et contiennent des pôles.

Ce sous-système est identifiable jusqu'à une convergence d'environ -65 dB avec un filtre FIR de 128 coefficients. La réponse impulsionnelle de ce sous-système et sa fonction de transfert sont montrées dans les figures 4.4 et 4.5 respectivement. Ceci montre que les filtres d'acquisitions peuvent être représentés très précisément par des filtres FIR. On remarque que la fréquence de coupure est de 50 Hz et la vitesse de coupure dans la bande d'atténuation est d'environ 10 dB/Hz. Puisque les deux filtres de conversions sont identiques, alors la vitesse de coupure de chaque filtre est environ 5 dB/Hz. Le bruit présent dans le système électronique limite la convergence. Le bruit peut aussi être observé dans la fonction de transfert identifiée, en particulier dans la bande entre 57 et 60 Hz où l'atténuation est plus de 70 dB.

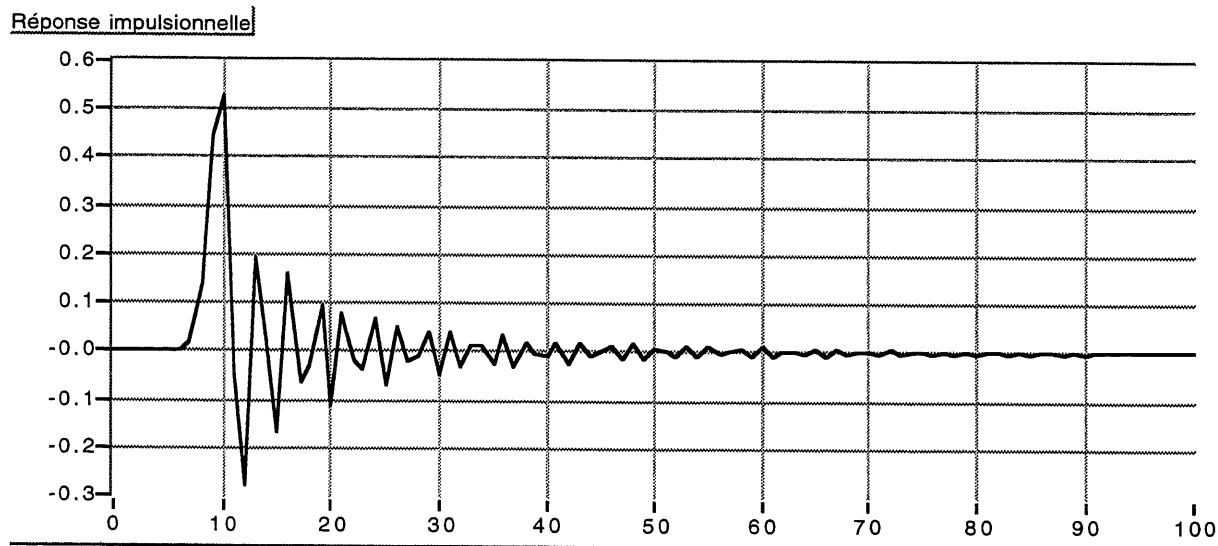


Figure 4.4 Réponse impulsionnelle de l'arrangement A/N-N/A en série



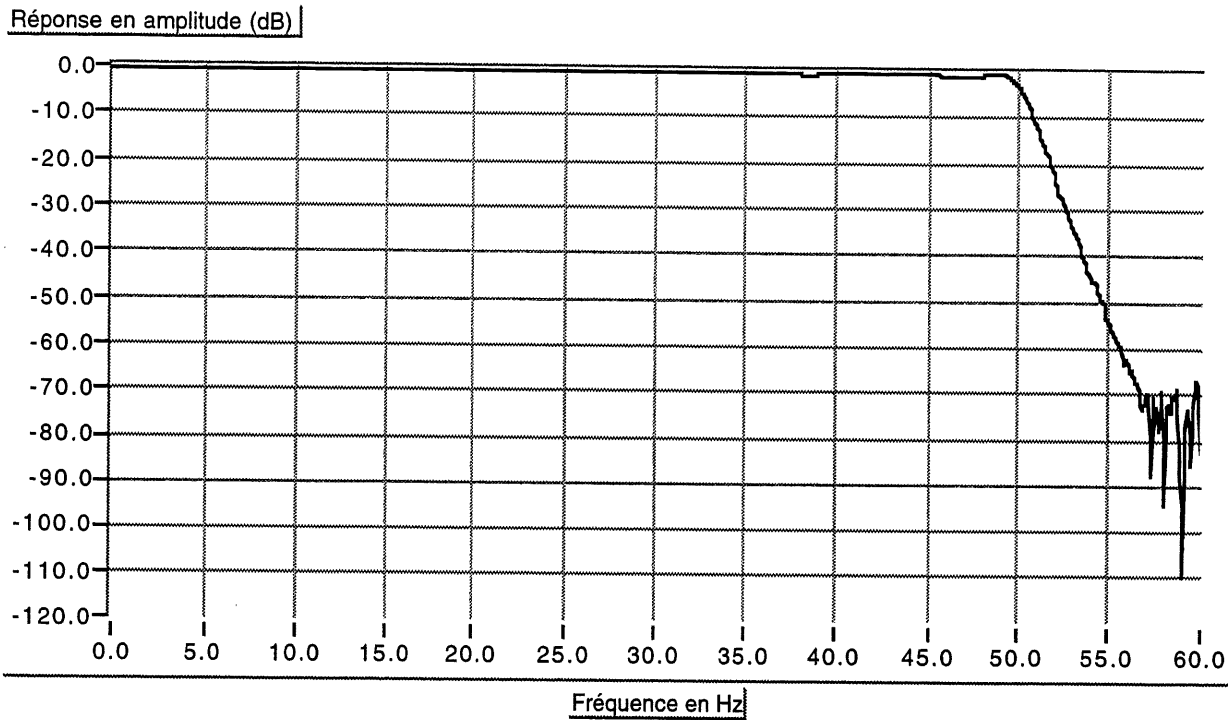


Figure 4.5 Fonction de transfert de l'arrangement A/N-N/A en série

Une fois que la fonction de transfert de l'arrangement A/N-N/A est identifiée, il est aussi intéressant d'obtenir la fonction de transfert et la réponse impulsionnelle de chaque convertisseur. Si  $G(j\omega)$  est la fonction de transfert identifiée, et que  $H(j\omega)$  est la fonction de transfert de chaque convertisseur (les deux convertisseurs sont identiques), alors :

$$G(j\omega) = H(j\omega).H(j\omega) \quad (4.8)$$

$$\text{Donc, } |H(j\omega)| = |G(j\omega)|^{1/2} \quad (4.9)$$

$$\text{et } \arg\{H(j\omega)\} = \frac{1}{2} \arg\{G(j\omega)\} \quad (4.10)$$

En calculant la phase de  $H(j\omega)$  avec l'équation (4.10), on doit être prudent. En effet, la phase de  $G(j\omega)$  prend des valeurs dans l'intervalle  $(-\pi, \pi]$ . Cette discontinuité de  $2\pi$  ne pose aucun problème dans la représentation de  $G(j\omega)$ . Mais si l'équation (4.10) est appliquée directement, la fonction  $\arg\{H(j\omega)\}$  comportera

des discontinuités de  $\pi$ , ce qui est inacceptable. Pour éviter ce problème, il est essentiel d'enlever les discontinuités de  $2\pi$  dans la fonction  $\arg\{G(j\omega)\}$  (dépliment de phase).

Une fois qu'on a calculé  $H(j\omega)$  avec les équations (4.9) et (4.10), on obtient la réponse impulsionnelle de chaque convertisseur en prenant la transformé inverse de Fourier. Les résultats sont donnés dans les figures 4.6 et 4.7.

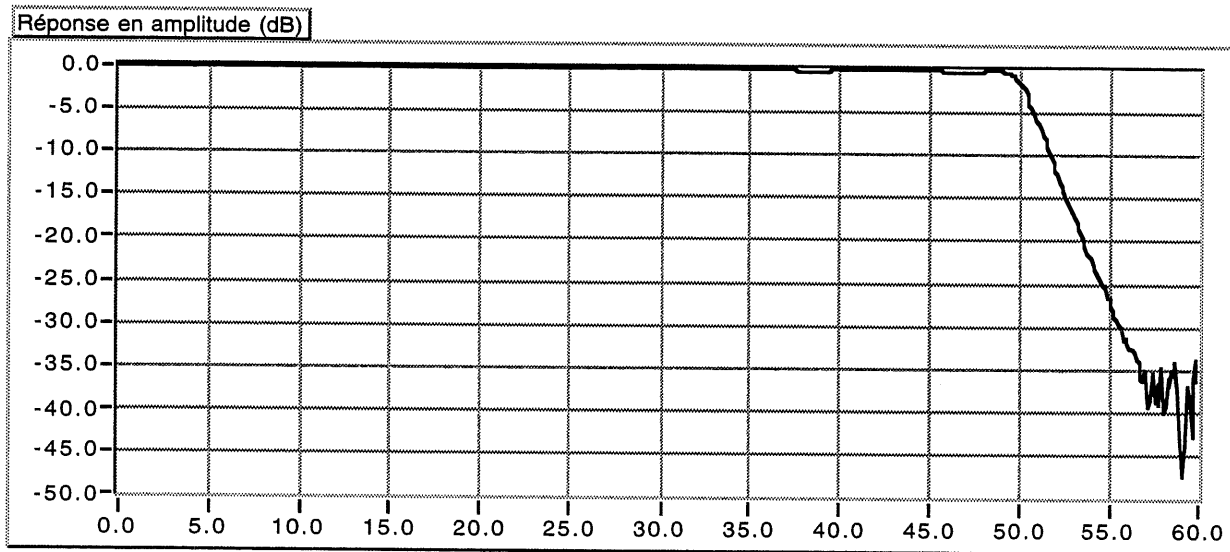


Figure 4.6 Fonction de transfert de chaque convertisseur

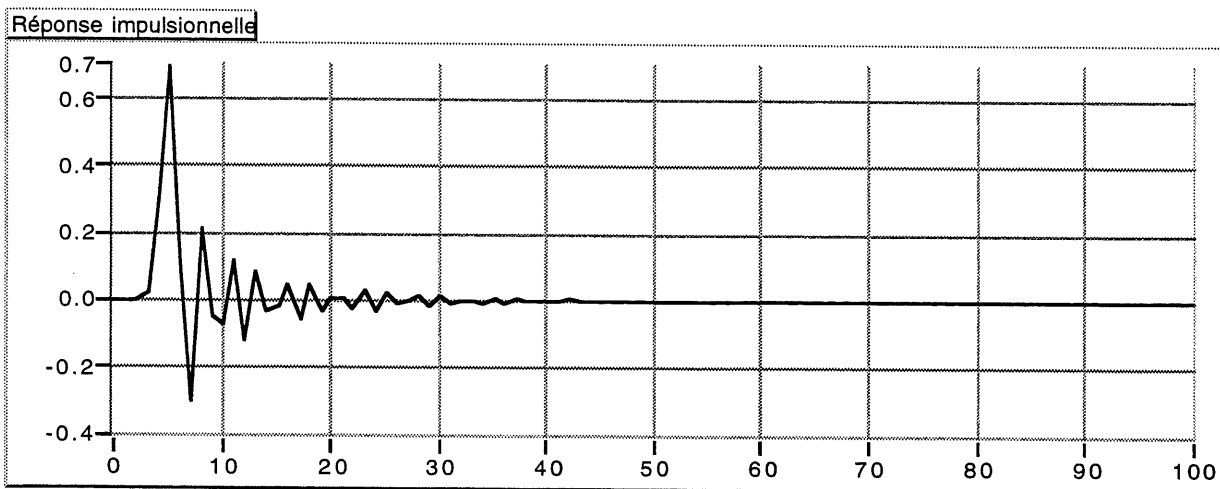


Figure 4.7 Réponse impulsionnelle de chaque convertisseur

#### 4.5 Identification du système par modèle ARMA adaptatif

On considère dans cette section une identification par modèle ARMA. On a vu que les filtres d'acquisition sont facilement représentés par des filtres FIR. Aussi, deux coefficients autorégressifs devraient en principe suffir pour représenter le pôle résonant à 15 Hz. Donc, un modèle ARMA aura la forme suivante :

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) - \sum_{j=1}^2 a_j y(n-j) \quad (4.11)$$

Deux méthodes peuvent être utilisées pour l'identification des systèmes par modèles ARMA: la méthode dite de l'erreur d'équation et la méthode d'erreur de sortie [19]. Dans la première méthode, qui est similaire à l'approche série-parallèle discutée dans la section 2.8, on utilise un modèle basé sur une équation non-récurssive pour faire l'identification :

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) - \sum_{j=1}^2 a_j d(n-j) \quad (4.12)$$

On remarque que la sortie désirée,  $d(n)$ , est retournée dans le modèle. On a ainsi un modèle avec deux signaux d'entrées,  $x(n)$  et  $d(n-1)$ , et une sortie  $y(n)$ . Le modèle comporte essentiellement deux filtres FIR agissant sur chacune des entrées. Cette méthode est illustrée dans la figure 4.8. Les coefficients de chaque filtre sont adaptés par l'algorithme de LMS pour filtre FIR.

L'autre méthode est basée sur une équation réursive, comme décrite par l'équation (4.11). Cette approche est similaire à l'identification en parallèle discutée dans la section 2.8. Les coefficients du modèle sont adaptés, soit par l'algorithme RPE (*recursive prediction error algorithm*), ou par l'algorithme PLR (*pseudolinear regression algorithm*) [19].

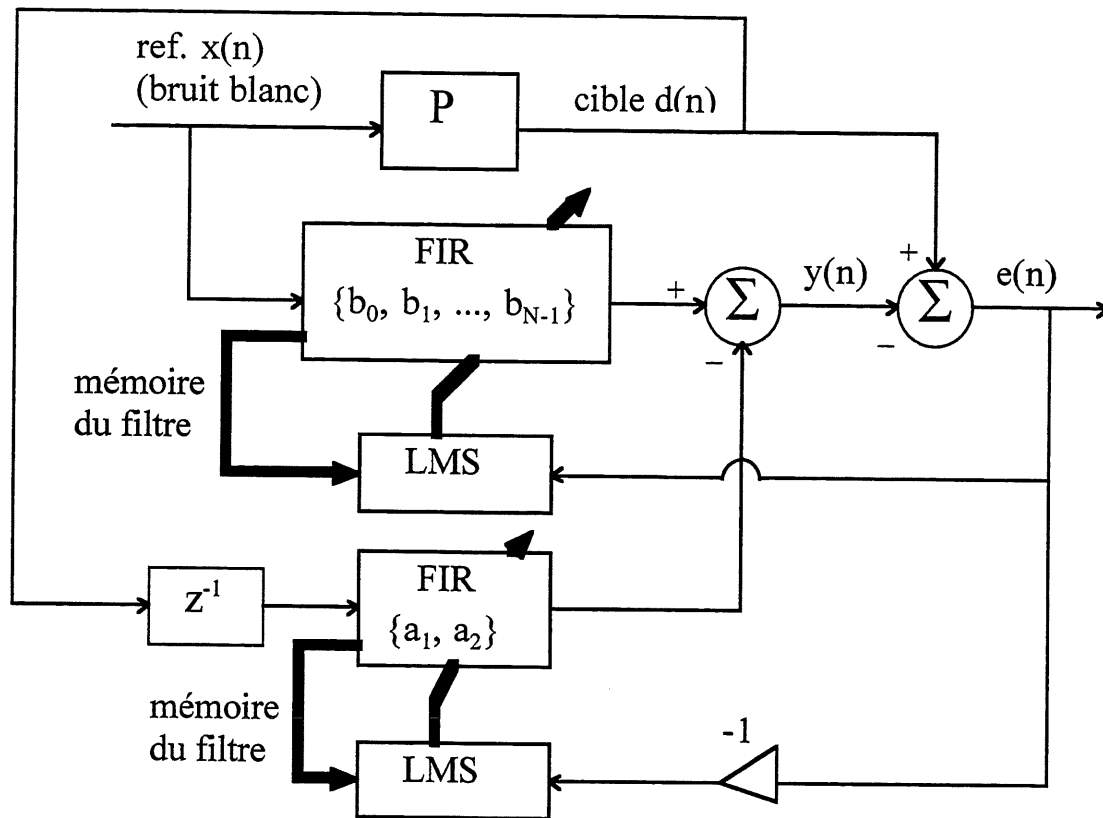


Figure 4.8 Identification d'un modèle ARMA par la méthode de l'erreur d'équation.

La première méthode a les avantages suivants. Premièrement, l'algorithme LMS pour les filtres FIR peut être appliqué directement. Deuxièmement, la convergence est assez rapide, et il n'y a pas le problème de minimums locaux. En plus, puisque le modèle est basé sur deux FIR, il est stable durant l'adaptation. Le désavantage de cette méthode, c'est qu'on utilise le signal  $d(n)$  comme une référence, et ce dernier est souvent contaminé par un bruit non-corrélé à l'entrée. Donc, les paramètres du modèle convergent vers une solution qui est biaisée par ce bruit.

La deuxième méthode, par contre, n'utilise pas  $d(n)$  comme une référence, et donc, ne donne pas une solution biaisée. Cependant, les algorithmes d'apprentissage sont plus complexes, la vitesse de convergence est relativement faible et le modèle peut devenir instable durant l'adaptation. Aussi, les paramètres peuvent converger vers un minimum local [19], ce qui donne une solution sous-optimale.

En pesant les pour et les contre, on choisit la première méthode, celle de l'erreur d'équation, pour les avantages qu'elle représente.

Dans une identification avec 256 coefficients pour la partie MA, on peut atteindre une convergence de  $-34 \pm 3$  dB. Autrement dit, l'amplitude de l'erreur d'identification, lorsque les paramètres ont convergé, est entre 1.4% et 2.8% de l'amplitude du signal cible. Pour vérifier que cette faible erreur d'identification n'est pas juste due à une adaptation à court terme des paramètres du modèle, on garde ces derniers fixes après l'adaptation et on observe le signal d'erreur. L'amplitude de l'erreur reste dans les bornes indiquées ci-dessus, ce qui implique que le modèle est fiable.

Les coefficients  $\{b_k\}$  identifiés sont montrés dans la figure 4.9. Les coefficients  $a_1$  et  $a_2$  identifiés sont -1.3992 et 0.9916 respectivement. On peut écrire la partie autorégressive de la fonction de transfert, c'est-à-dire  $1 + a_1 z^{-1} + a_2 z^{-2}$ , dans la forme suivante :

$$1 - 2\gamma \cos\left(\frac{2\pi f_0}{f_s}\right) z^{-1} + \gamma^2 z^{-2} \quad (4.13)$$

où  $f_0$  et  $\gamma$  sont la fréquence et le coefficient d'amortissement respectivement du pôle résonant et  $f_s$  est la fréquence d'échantillonnage. En calculant  $f_0$  et  $\gamma$ , utilisant  $f_s = 120.0$  Hz et les valeurs  $a_1$  et  $a_2$  identifiées, on obtient  $f_0 = 15.12$  Hz et  $\gamma = 0.996$ . Les spectres du signal cible et d'erreur, ainsi que la fonction de transfert identifiée sont montrés dans la figure 4.10.

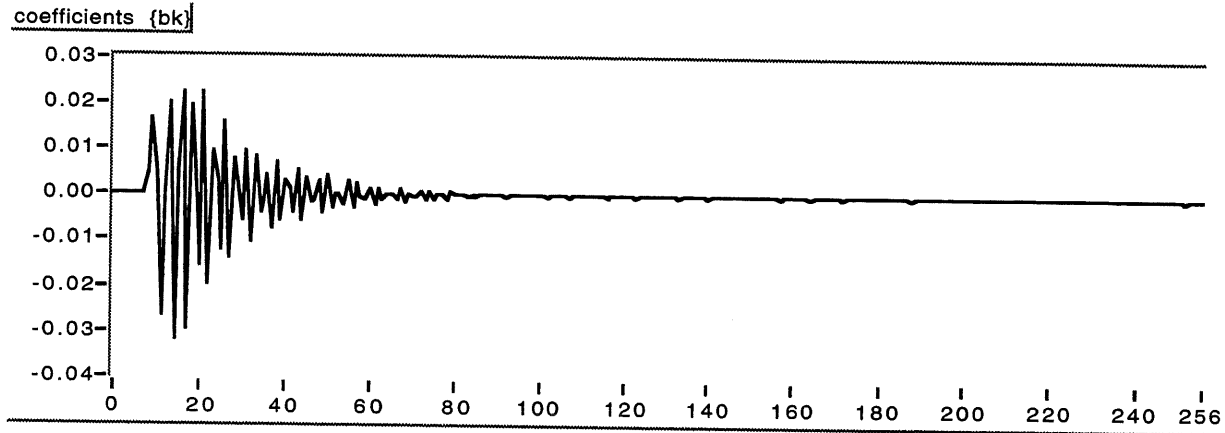


Figure 4.9 Coefficients  $\{b_k\}$  pour le modèle ARMA identifié

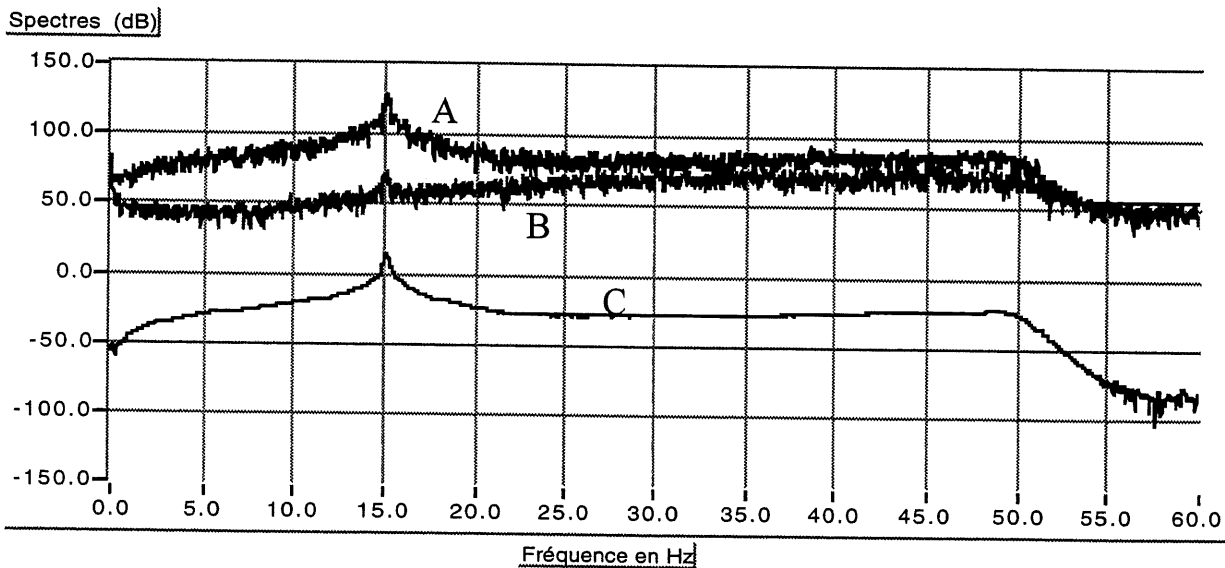


Figure 4.10 Spectres du signal cible (A) et d'erreur (B) et fonction de transfert identifiée (C)

En examinant la figure 4.9, on remarque qu'on aurait pu utiliser seulement 128 coefficients ou moins pour la partie MA. On fait les observations suivantes sur la figure 4.10. Premièrement, on remarque que l'énergie de l'erreur est plus importante aux fréquences plus grandes que la fréquence de résonance. Deuxièmement, l'atténuation dans la bande de rejet (50 à 60 Hz) de la fonction de transfert identifiée est deux fois plus importante que l'atténuation observée dans le spectre du signal cible. Ceci est une conséquence de la non-linéarité présente dans le sous-système mécanique entre les deux filtres de conversions.

Il est aussi intéressant de calculer la réponse impulsionnelle du système avec les coefficients  $\{a_k\}$  et  $\{b_k\}$  identifiés et de la comparer avec la réponse impulsionnelle mesurée directement sur le système. Pour avoir des meilleurs résultats, on utilise une impulsion de taille 30,000. Les réponses sont illustrées dans les figures 4.11 et 4.12. La différence entre les deux réponses est aussi montrée dans la figure 4.13.

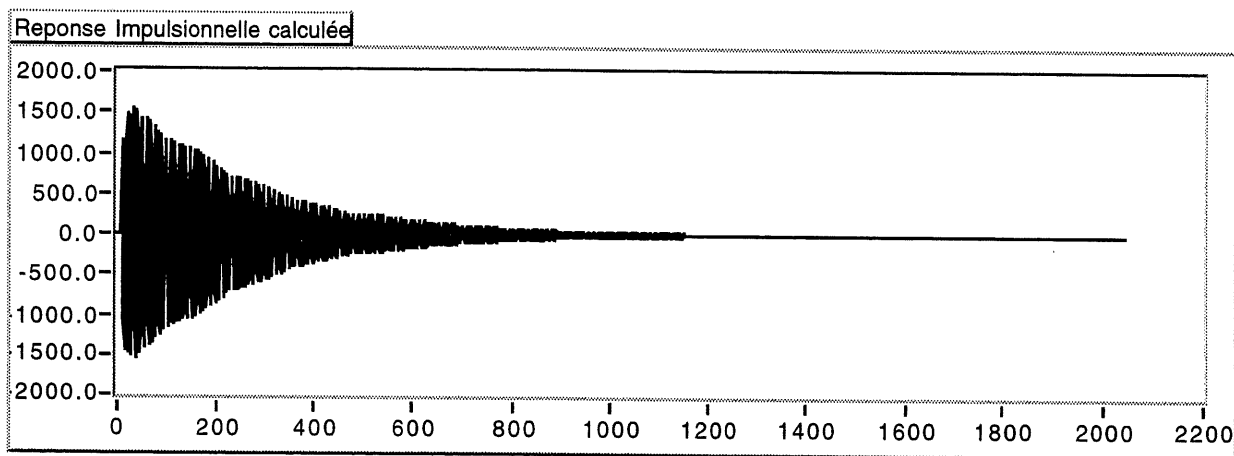


Figure 4.11 Réponse impulsionnelle calculée

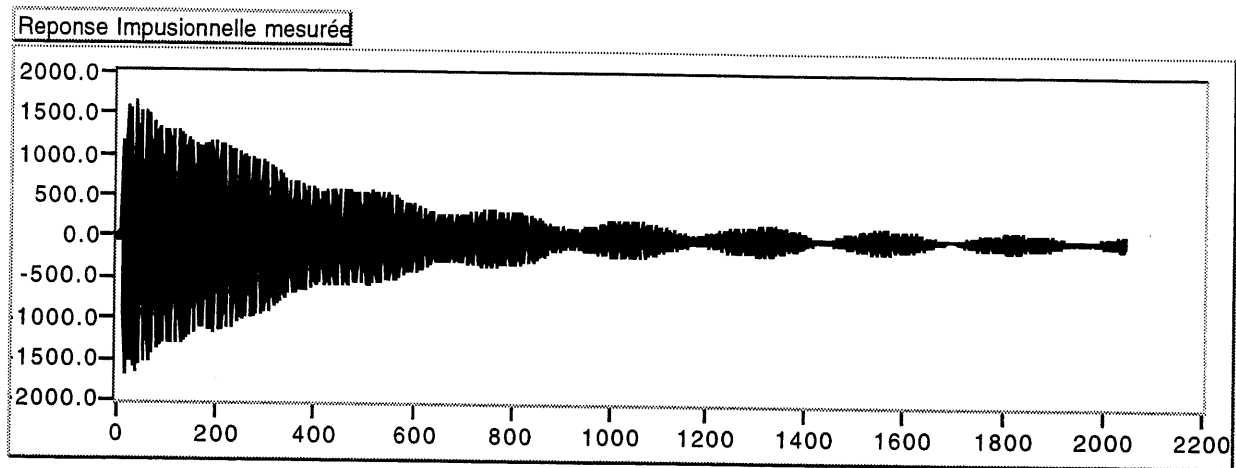


Figure 4.12 Réponse impulsionnelle mesurée

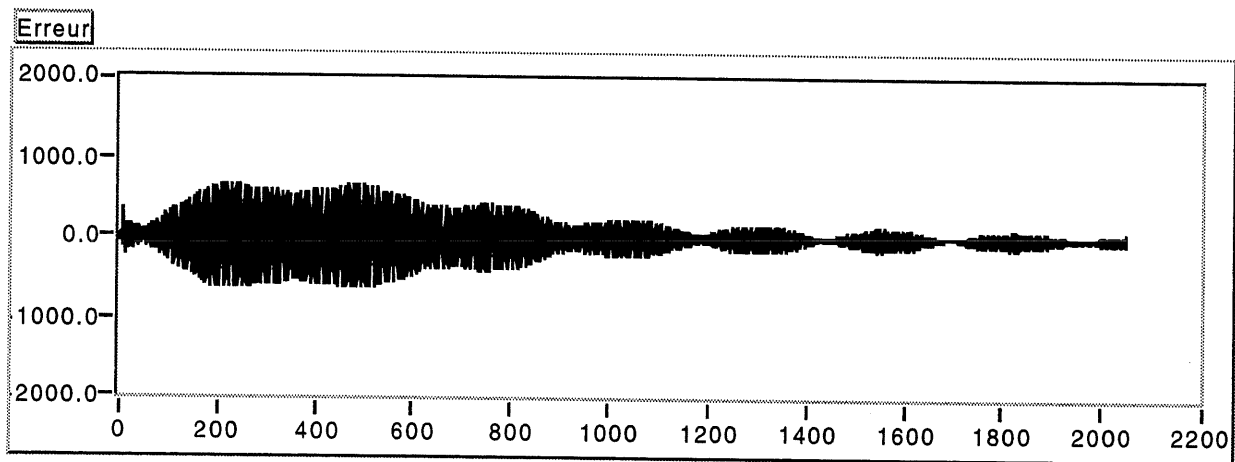


Figure 4.13 Différence entre les réponses

En examinant la figure 4.13, on voit que la différence entre les deux réponses est très importante. Cela peut s'expliquer comme suit. D'abord, on remarque un battement à une fréquence de 0.46 Hz dans la réponse impulsionnelle mesurée. Cela veut dire que la réponse contient deux fréquences dominantes très proches (à une séparation de 0.46 Hz). Alors, il se peut que le deuxième ordre pour la partie autoregressive du modèle n'est pas adéquat. Cet effet peut aussi être dû à la non-linéarité du système. Ensuite, le fait que la réponse impulsionnelle de la figure 4.11 est calculée par un modèle récursif, c'est-à-dire que la sortie du modèle est retournée à l'entrée, alors la sortie du modèle dérive par rapport à la



sortie du système physique. L'erreur sur chaque coefficient identifié contribue à une erreur à la sortie du modèle, et cette erreur est retournée à l'entrée. Bien que pour les premiers cent échantillons l'effet n'est pas trop important, la dérive de la sortie du modèle par la suite est plutôt importante. Ceci est observable en faisant un agrandissement sur les réponses, telles que montrées dans les figures 4.14(a) et 4.14(b).

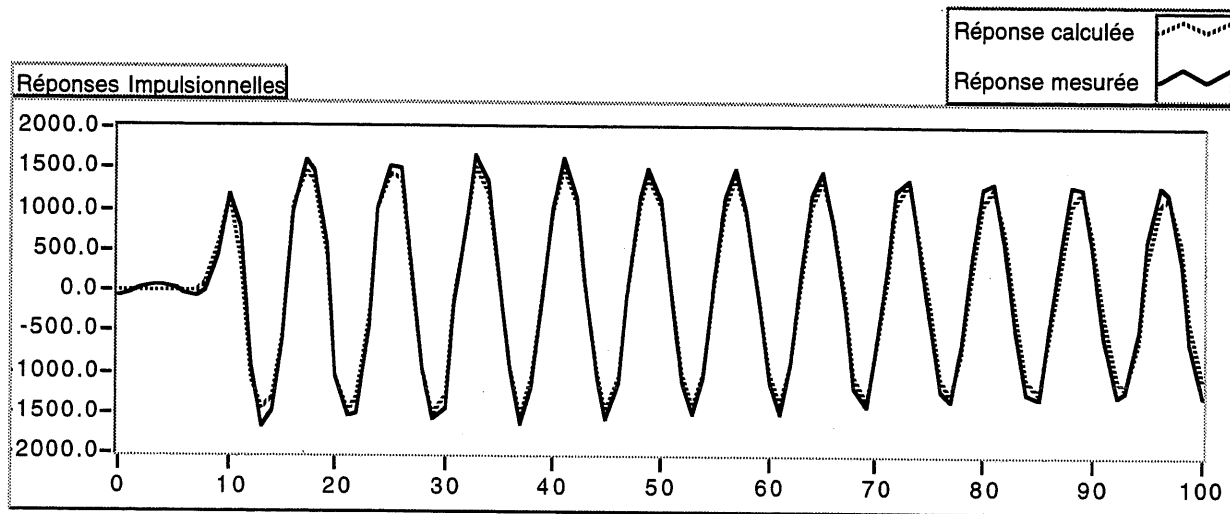


Figure 4.14(a) Agrandissement sur les premiers 100 échantillons

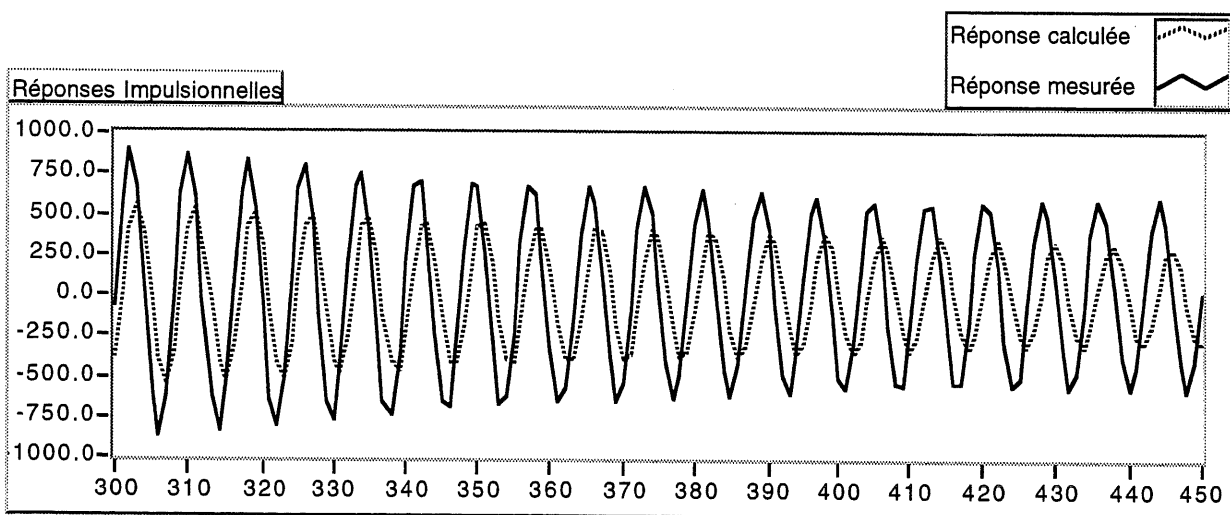


Figure 4.14(b) Agrandissement sur les échantillons entre 300 et 450

La différence entre les deux réponses est aussi due en partie à la présence du bruit dans la réponse impulsionnelle mesurée.

#### **4.6 Conclusion**

On a vu dans ce chapitre qu'une identification du système basée sur un modèle MA nécessite beaucoup trop de coefficients, et ne donne pas une convergence satisfaisante. Cependant, cette identification nous donne des indications pour le choix d'un modèle ARMA.

On a aussi vu que les filtres de conversions sont identifiables très précisément (jusqu'à environ -65 dB d'erreur) par un modèle MA avec peu de coefficients. Cela nous donne aussi une limite supérieure à la performance d'une identification non-linéaire du système, c'est-à-dire qu'on ne pourrait pas faire mieux que -65 dB.

L'identification par un modèle ARMA avec seulement deux coefficients pour la partie AR, et 256 coefficients pour la partie MA, donne une convergence bien meilleure que le modèle MA. La convergence est à -34 dB en moyenne. Donc, lors d'une identification non-linéaire, il s'agirait de faire une meilleure performance que cela.

## 5. IDENTIFICATION NON-LINÉAIRE PAR RÉSEAUX DE NEURONES

### 5.1 Introduction

Dans ce présent chapitre, on présente les formulations théoriques pour l'identification des systèmes par des réseaux de neurones. On commence par présenter des modèles généraux des systèmes non-linéaires. Ensuite, on présente un réseau de neurones de deux couches et un algorithme rapide pour son apprentissage. On propose aussi une amélioration sur cet algorithme. Finalement, on présente les méthodes d'identification pour chacun des quatre modèles et la méthode de vérification de ces modèles.

### 5.2 Modèles d'identification non-linéaires

Il existe beaucoup de modèles pour la représentation des systèmes non-linéaires. On peut avoir des modèles très structurés qui sont basés sur une connaissance physique du système qu'on cherche à identifier. On peut par contre avoir des modèles généraux qui ont pour but de représenter la relation entrée-sortie du système, sans toutefois avoir une interprétation physique de ce dernier. Ces modèles ont l'avantage d'être applicables pour une plus grande gamme de systèmes. Ce sont ces types de modèles qu'on utilise dans cette étude.

Si on suppose que la sortie  $y(n)$  du système ne dépend que de l'entrée  $x$ , alors le modèle MA non-linéaire suivant peut être utilisé:

$$y(n) = f[x(n), x(n-1), x(n-2), \dots, x(n-N)] \quad (5.1)$$

où  $N$  est l'ordre de mémoire du système et  $f$  est une fonction non-linéaire. Ce modèle ne peut être utilisé que pour des systèmes à mémoire finie. Dans beaucoup de cas ce n'est pas un modèle adéquat.

Si la sortie  $y(n)$  du système dépend de l'entrée  $x$  ainsi que des sorties précédentes  $y(n-1)$ ,  $y(n-2)$ ,...etc., alors on utilise des modèles ARMA non-linéaires. On reprend les quatre modèles pour la représentation des systèmes SISO non-linéaires donnés dans [10].

$$\text{Modèle I : } y(n) = \sum_{i=1}^M a_i y(n-i) + g[x(n), x(n-1), \dots, x(n-N)] \quad (5.2)$$

$$\text{Modèle II :} \quad (5.3)$$

$$y(n) = f[y(n-1), y(n-2), \dots, y(n-M)] + \sum_{i=0}^N b_i x(n-i)$$

$$\text{Modèle III :} \quad (5.4)$$

$$y(n) = f[y(n-1), y(n-2), \dots, y(n-M)] + g[x(n), x(n-1), \dots, x(n-N)]$$

$$\text{Modèle IV :} \quad (5.5)$$

$$y(n) = f[y(n-1), y(n-2), \dots, y(n-M), x(n), x(n-1), \dots, x(n-N)]$$

où  $[x(n), y(n)]$  représente la paire entrée-sortie du système SISO au temps  $n$ , et  $f(.)$  et  $g(.)$  représentent des fonctions non-linéaires générales.

Notons que le modèle III est plus général que les deux premiers. Aussi, le modèle IV est le plus général de tous. Cependant, une interprétation devient plus difficile avec plus de généralités.

### 5.3 Réseau de neurones feedforward de deux couches

Dans cette étude, on n'utilise que des réseaux de neurones de deux couches seulement. De plus, la couche de sortie est linéaire, c'est-à-dire sans fonction d'activation, et il n'y a qu'une seule sortie. Ce type de réseau est illustré dans la figure 5.1.

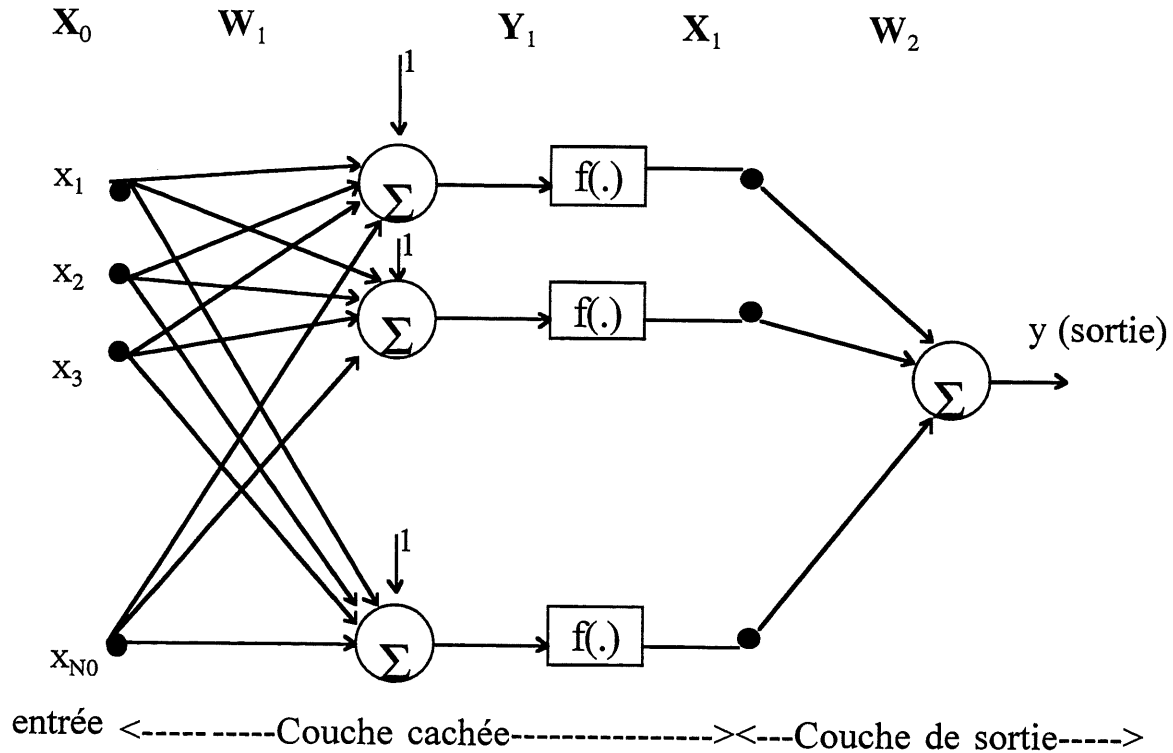


Figure 5.1 Réseau de neurones de deux couches

L'entrée du réseau est le vecteur  $\mathbf{X}_0$ . La première couche (couche cachée) du réseau se compose de la matrice des poids  $\mathbf{W}_1$ , les additionneurs, et les fonctions d'activations  $f(\cdot)$ . À la sortie de la première couche, on a le vecteur  $\mathbf{X}_1$ . La sortie  $y$  du réseau est une somme pondérée (par  $\mathbf{W}_2$ ) des sorties de la couche cachée. Si  $N_0$  est le nombre d'entrée du réseau et  $N_1$  est le nombre de neurones dans la couche cachée, alors:

$$\mathbf{X}_0 = [x_1, x_2, \dots, x_{N_0}]^T \quad (5.6)$$

$$\mathbf{X}_0^* = [1, x_1, x_2, \dots, x_{N_0}]^T \quad (5.7)$$

$$\mathbf{Y}_1 = \mathbf{W}_1 \cdot \mathbf{X}_0^* \quad (5.8)$$

$$\mathbf{X}_1 = [f(Y_{1,1}), f(Y_{1,2}), \dots, f(Y_{1,N_1})]^T \quad (5.9)$$

$$y = \mathbf{W}_2^T \cdot \mathbf{X}_1 \quad (5.10)$$

Les fonctions d'activations  $f(\cdot)$  peuvent être sigmoïdes ou de type tangent hyperbolique. Ce sont ces dernières qu'on utilise dans ce travail. C'est-à-dire que:

$$Y_{1,i} = \tanh(\beta X_{1,i}) \quad (5.11)$$

où  $\beta$  est arbitrairement choisis égal à 1.

Il a été démontré qu'un tel réseau peut bien estimer n'importe quelle fonction continue et absolument-intégrable,  $g(x_1, x_2, \dots, x_{N_0})$ , si le nombre de neurones  $N_1$  est assez grand [2]. C'est cette propriété qui est exploitée pour la modélisation des systèmes non-linéaires dynamiques. On utilise une ligne à retard (*tapped delay line*) pour convertir un signal temporel en signal spatial à l'entrée du réseau. Pour modéliser un système non-linéaire dynamique avec  $(u(n), y(n))$  comme la paire entrée-sortie, on peut définir  $\mathbf{X}_0$  comme  $[u(n), u(n-1), \dots, u(n-N)]$  ou  $[y(n-1), y(n-2), \dots, y(n-M)]$  ou même  $[u(n), u(n-1), \dots, u(n-N), y(n-1), \dots, y(n-M)]$ , dépendant du modèle choisi.

Il faut aussi noter qu'en pratique les entrées dans le vecteur  $\mathbf{X}_0$  sont normalisées pour qu'elles soient dans l'intervalle  $[-1, 1]$ . Cela nous permet d'avoir une matrice  $\mathbf{W}_1$  plus facile à initialiser et à contrôler.

## 5.4 Algorithme de rétropropagation améliorée [Scalero, 1992]

Le principe d'identification par réseau de neurones artificielle (RNA) est illustré dans la figure 5.2.

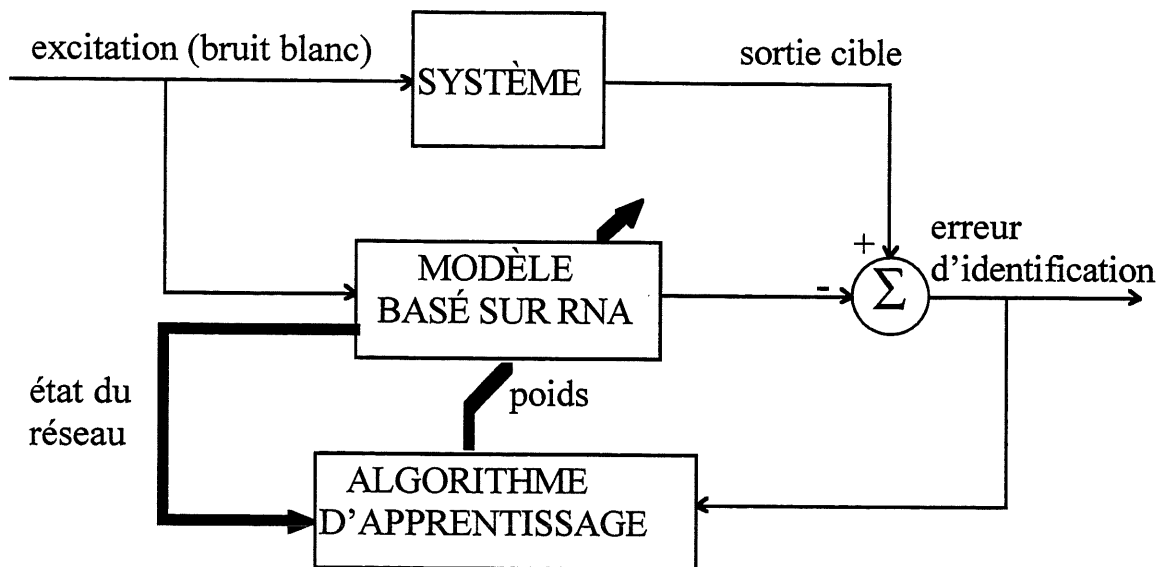


Figure 5.2 Identification par réseau de neurones

L'algorithme de rétropropagation améliorée (*enhanced backpropagation*) [15] pour l'apprentissage d'un réseau de neurones de type *feedforward* est décrit dans cette section. Cet algorithme est beaucoup plus rapide que l'algorithme de rétropropagation standard. Dans l'article original [15], l'algorithme est décrit pour un réseau de neurones général de plusieurs couches. Ici on présente l'algorithme adapté pour le réseau décrit dans la section précédente.

Voici l'algorithme :

### 1) Initialisations

Avant de commencer l'apprentissage, on initialise les matrices de poids  $W_1$  et  $W_2$  et les matrices de corrélation inverses  $R_1^{-1}$  et  $R_2^{-1}$  comme suit :

$W_1$  est une matrice de dimension  $N_1 \times (N_0 + 1)$  avec les entrées aléatoirement choisies dans l'intervalle  $\left(-\frac{1}{3\sqrt{N_0+1}}, \frac{1}{3\sqrt{N_0+1}}\right)$ .

$W_2$  est un vecteur de  $N_1$  entrées aléatoirement choisies dans l'intervalle  $\left(-\frac{1}{3\sqrt{N_1}}, \frac{1}{3\sqrt{N_1}}\right)$ .

$R_1^{-1}$  est une matrice de dimension  $(N_0 + 1) \times (N_0 + 1)$  et de valeur  $R_1^{-1} = \delta^{-1}I_{N_0+1}$  où  $\delta$  est une petite valeur positive ( $\ll 1$ ) et  $I_{N_0+1}$  est la matrice d'identité de même dimensions que  $R_1^{-1}$ .

$R_2^{-1}$  est une matrice de dimension  $N_1 \times N_1$  et de valeur  $R_2^{-1} = \delta^{-1}I_{N_1}$  où  $I_{N_1}$  est la matrice d'identité de même dimensions que  $R_2^{-1}$ .

Ensuite, pour chaque itération  $n$ , c'est-à-dire pour chaque association entrée-sortie  $(x(n), d(n))$ , on suit les étapes suivantes :

2) Les vecteurs  $X_0$  et  $X_0^*$  sont formés à partir de  $x(n)$  et/ou la sortie désirée  $d(n)$  et les vecteurs  $Y_1$  et  $X_1$ , ainsi que la sortie  $y(n)$  du réseau sont calculés en utilisant les équations dans la section précédente.

3) Pour chaque couche, les vecteurs de gains de Kalman,  $k_1$  et  $k_2$  sont calculés et les matrices d'autocorrélation inverses sont mises à jour d'après les équations suivantes :

$$k_1 = \frac{R_1^{-1} X_0^*}{b + X_0^{*T} R_1^{-1} X_0} \quad (\text{couche cachée}) \quad (5.12)$$



$$\mathbf{k}_2 = \frac{\mathbf{R}_2^{-1} \mathbf{X}_1}{b + \mathbf{X}_1^T \mathbf{R}_2^{-1} \mathbf{X}_1} \quad (\text{couche de sortie}) \quad (5.13)$$

$$\mathbf{R}_1^{-1}(n+1) = \left[ \mathbf{R}_1^{-1}(n) - \mathbf{k}_1 \mathbf{X}_0^{*T} \mathbf{R}_1^{-1}(n) \right] b^{-1} \quad (\text{couche cachée}) \quad (5.14)$$

$$\mathbf{R}_2^{-1}(n+1) = \left[ \mathbf{R}_2^{-1}(n) - \mathbf{k}_2 \mathbf{X}_1^T \mathbf{R}_2^{-1}(n) \right] b^{-1} \quad (\text{couche de sortie}) \quad (5.15)$$

où  $b$  est un facteur d'oubli proche de l'unité mais plus faible.

4) Le signal d'erreur  $e(n)$  à la sortie du réseau est évalué et rétropropagé à la sortie de la couche cachée.

$$e(n) = d(n) - y(n) \quad (5.16)$$

$$\mathbf{e}_1 = \mathbf{z} \otimes (\mathbf{W}_2 \cdot e(n)) \quad (5.17)$$

$$\text{où } \mathbf{z} = [f'(Y_{1,1}), f'(Y_{1,2}), \dots, f'(Y_{1,N1})] \quad (5.18)$$

$$f'(a) = \beta \operatorname{sech}^2(\beta \cdot a) \quad (5.19)$$

et  $\otimes$  représente le produit élément par élément sur deux vecteurs.

5) Les poids dans  $\mathbf{W}_1$  et  $\mathbf{W}_2$  sont mis à jour.

$$\mathbf{W}_2(n+1) = \mathbf{W}_2(n) + \mu_2 e(n) \mathbf{k}_2 \quad (5.20)$$

$$\mathbf{W}_1(n+1) = \mathbf{W}_1(n) + \mu_1 \mathbf{e}_1 \mathbf{k}_1^T \quad (5.21)$$

où  $\mu_1$  et  $\mu_2$  sont les pas d'adaptation. Des valeurs typiques sont  $\mu_1 = 1/4$  et  $\mu_2 = 1/2$ .

6) Si la puissance du signal d'erreur  $e(n)$  est assez petite, on arrête l'adaptation. Sinon on retourne à l'étape 2 pour l'itération suivante.

Notons que les matrices de poids sont mises à jour à chaque échantillon des signaux d'entrée et de sortie, à l'aide d'un algorithme itératif rapide. Cela représente un certain avantage par rapport aux algorithmes qui fonctionnent avec la mise à jour périodique des paramètres.

### **5.5 Modification sur l'algorithme de rétropropagation améliorée**

L'algorithme de rétropropagation améliorée, décrit dans la section précédente, a été implanté avec le logiciel LabVIEW et testé avec des exemples simples. Dans ces exemples, on essaye de représenter des fonctions unidimensionnelles telles que  $x^2$ ,  $x^3$  et  $\sin x$  par des réseaux de neurones de deux couches. Dans un autre exemple on essaye d'identifier un système décrit par l'équation  $y(n) = y(n-1)/[1 + y(n-1)^3] + x^3(n)$  par un réseau. Dans ces tests, on s'aperçoit que si les pas d'adaptation  $\mu_1$  et  $\mu_2$  sont trop grands, le réseau a tendance à entrer dans des minimums locaux (et y rester longtemps) dû à la saturation excessive des neurones. Par contre, si  $\mu_1$  et  $\mu_2$  sont trop petits la vitesse de convergence diminue.

Il est aussi fait mention dans [3] que la protection contre les changements excessifs des poids est un des objectifs à viser dans l'apprentissage d'un réseau de neurones, car cela mène à la saturation excessive des neurones. Cependant, les formulations développées directement avec cette contrainte [3] sont très complexes à implanter.

On développe dans cette section une technique plus simple qui permet une protection efficace contre la saturation excessive des neurones. Cette technique et son développement sont originaux. De plus, puisque l'apprentissage

d'un réseau de neurones est de nature statistique, la technique est développée sur des bases statistiques.

Considérons un neurone individuel dans la couche cachée. Le signal d'entrée  $Y$  pour la fonction d'activation est donné par :

$$Y = \sum_{i=1}^{N_0} w_i x_i \quad (5.22)$$

où  $w_i$  est le poids de l'entrée  $x_i$  au neurone considéré et  $N_0$  est le nombre d'entrée du réseau.

Supposons que les  $w_i$  sont fixes et que les entrées  $x_i$  sont uniformément distribuées dans l'intervalle  $[-1,1]$ . Afin de simplifier la formulation, on suppose aussi que les entrées  $x_i$  sont indépendantes l'une de l'autre. Alors,

$$E\{Y\} = 0$$

$$\text{et } \text{Var}\{Y\} = \text{Var}\left\{\sum_{i=1}^{N_0} w_i x_i\right\} \quad (5.23)$$

$$= \sigma_x^2 \sum_{i=1}^{N_0} w_i^2 \quad (5.24)$$

Sous l'hypothèse que  $Y$  est une somme de variables indépendantes, ce qui suppose la blancheur du signal d'excitation du réseaux, le théorème central-limite nous dit que pour  $N_0$  assez grand, la distribution de  $Y$  se rapproche d'une distribution Gaussienne. C'est-à-dire que :

$$Y \sim N(0, \sigma_Y^2) \quad (5.25)$$

On sait aussi que pour une distribution Gaussienne, on est à 99.7% sûr que la variable  $Y$  prend des valeurs dans l'intervalle  $(-3\sigma_Y, 3\sigma_Y)$ . C'est-à-dire que c'est plutôt sûr que :

$$|Y| < 3 \sigma_Y \quad (5.26)$$

On établit maintenant une limite pour le taux de saturation d'un neurone. Si on utilise la fonction  $f(Y) = \tanh(\beta Y)$  comme fonction d'activation, le neurone est saturé lorsque sa sortie se rapproche de bornes (-1 ou 1) de la fonction hyperbolique. Puisque l'algorithme d'adaptation fait usage de  $f'(Y)$  au lieu de  $f(Y)$ , on place la contrainte sur ce premier. Puisque la fonction  $f'(Y) = \beta \operatorname{sech}^2(\beta Y)$  prend des valeurs entre 0 et  $\beta$  et que la valeur 0 représente une saturation totale du neurone, on fixe une limite minimale à  $f'(Y)$ , soit  $\varepsilon\beta$ . Une valeur de 0.9 pour  $\varepsilon$  implique qu'on se limite à la région très linéaire de la fonction d'activation et une valeur de 0 veut dire qu'il n'y pas de contrainte. Durant l'apprentissage, une valeur typique pour  $\varepsilon$  est de 0.1 qui a pour but d'offrir une opération assez non-linéaire sans trop saturer le neurone. Donc, pour un neurone non-saturable, on a :

$$|Y| \leq \frac{1}{\beta} \operatorname{sech}^{-1}(\sqrt{\varepsilon}) \quad (5.27)$$

En combinant les inégalités dans les équations (5.26) et (5.27), on obtient la contrainte qui assure la non-saturation des neurones. C'est-à-dire que :

$$3\sigma_Y \leq \frac{1}{\beta} \operatorname{sech}^{-1} \sqrt{\varepsilon} \quad (5.28)$$

En utilisant l'équation (5.24) pour  $\sigma_Y$ , on obtient :

$$3\sigma_x \sqrt{\sum_{i=1}^{N0} w_i^2} \leq \frac{1}{\beta} \operatorname{sech}^{-1} \sqrt{\varepsilon} \quad (5.29)$$

Donc, la contrainte qu'il nous faut imposer sur les poids pour éviter la saturation excessive des neurones est explicitement écrite comme suit :

$$\sum_{i=1}^{N0} w_i^2 \leq \left( \frac{\text{sech}^{-1} \sqrt{\epsilon}}{3 \beta \sigma_x} \right)^2 \quad (5.30)$$

On tient à remarquer que, lors de la dérivation de l'équation (5.30), on suppose que les distributions des entrées sont indépendantes. En réalité, ce n'est souvent pas le cas. Si les entrées du réseau proviennent d'une ligne à retard agissant sur un signal blanc, alors la supposition est tout à fait valide. Si par contre, les entrées du réseau proviennent d'une ligne à retard qui contient les échantillons de la sortie d'un système, alors la supposition peut être erronée. En principe, si le signal d'entrée n'est pas blanc, on devrait tenir compte de sa coloration spectrale pour calculer la variance du signal à l'entrée de chaque fonction d'activation. Toutefois, l'hypothèse de blancheur permet une approximation suffisante pour le but recherché. L'erreur encourue ne justifie pas l'augmentation de complexité découlant de ce raffinement.

Dans ce nouvel algorithme, cette contrainte est appliquée de la manière suivante. Les étapes 1 à 4 restent les mêmes que dans la section précédente. L'étape 5 est modifiée de la manière suivante. Les poids sont d'abord modifiés d'après les équations 5.20 et 5.21. Ensuite, on vérifie pour chaque neurone si la contrainte de l'équation 5.30 est respectée. Si elle l'est, on passe à l'étape suivante. Sinon, les poids pour les neurones saturés sont réinitialisés aléatoirement comme lors de l'initialisation. Cette réinitialisation aléatoire peut paraître un peu brutale, mais c'est dans l'intérêt de conserver une nature aléatoire à l'apprentissage. Évidemment, il y a encore des améliorations à faire sur la façon dont on corrige les poids d'un neurone excessivement saturé.

Des tests faits avec les mêmes exemples précédents démontrent que pour les mêmes paramètres d'adaptation ( $\mu_1, \mu_2$ , etc.), cette modification améliore la vitesse d'apprentissage de l'algorithme décrit dans la section précédente, surtout si les valeurs de  $\mu_1$  et  $\mu_2$  sont assez grandes. Il semble aussi que les minimums locaux soient assez bien évités avec cette modification. Cette technique est donc utilisée dans tous les apprentissages de réseau de ce travail.

L'équation (5.30) est aussi utile pour l'initialisation des poids. En pratique, il est préférable d'initialiser les poids avec des valeurs assez petites pour que tous les neurones opèrent dans leur région linéaire au départ.

Supposons qu'on initialise les poids aléatoirement par une distribution uniforme dans l'intervalle  $(-a, a)$ . Alors, on obtient :

$$E\{\mathbf{w}_i^2\} = a^2 / 3 \quad (5.31)$$

En prenant l'espérance sur l'équation 5.30, on obtient :

$$N_0 \cdot E\{\mathbf{w}_i^2\} \leq \left( \frac{\text{sech}^{-1} \sqrt{\varepsilon}}{3 \beta \sigma_x} \right)^2 \quad (5.32)$$

En combinant les équations 5.31 et 5.32, on a :

$$a \leq \sqrt{\frac{3}{N_0}} \cdot \frac{\text{sech}^{-1} \sqrt{\varepsilon}}{3 \beta \sigma_x} \quad (5.33)$$

En utilisant  $\varepsilon = 0.9$  (très linéaire),  $\beta = 1.0$ , et  $\sigma_x = 1/\sqrt{3}$ , on obtient :

$$a \approx \frac{1}{3\sqrt{N_0}} \quad (5.34)$$

C'est-à-dire qu'on choisit les poids aléatoirement dans l'intervalle  $(-\frac{1}{3\sqrt{N_0}}, \frac{1}{3\sqrt{N_0}})$  lors de l'initialisation.

## 5.6 Méthodes pour les identifications

Dans les sections précédentes, l'algorithme pour l'adaptation d'un réseau de neurones de deux couches a été présenté. Cependant, pour faire l'identification des modèles dans les équations 5.2 à 5.5, il y a plusieurs précautions à prendre. L'adaptation est faite avec l'algorithme pour réseau *feedforward* bien que les modèles soient récurrents. Cela est possible en faisant des identifications série-parallèles (voir section 2.8). Puisqu'on suppose que les entrées des réseaux varient dans l'intervalle  $[-1,1]$ , une normalisation basée sur une connaissance des amplitudes des signaux d'entrée et de sortie est indispensable. Pour les trois premiers modèles, on adapte simultanément deux structures en parallèle: soit un réseau et un filtre linéaire ou deux réseaux. Dans ces cas, l'erreur d'identification à la sortie du modèle est multipliée par des facteurs (dépendant du modèle) pour obtenir l'erreur à la sortie de chaque branche parallèle. De plus, les filtres linéaires sont adaptés d'après l'algorithme LMS normalisé (voir le chapitre précédent), tandis que les réseaux sont adaptés d'après l'algorithme présenté dans les sections 5.4 et 5.5. Le signal de référence  $x(n)$  est un bruit blanc pour s'assurer un apprentissage en large bande.

Les structures pour l'identification de chaque modèle sont présentées dans les figures 5.3 à 5.6. Les significations des abréviations utilisées dans ces figures sont données plus bas.

$A_d$	=	amplitude moyenne du signal de sortie du système
$A_x$	=	amplitude du signal de référence
$d_{NN}(n)$	=	signal cible du réseau de neurones
$A_{dNN}$	=	amplitude moyenne du signal cible du réseau de neurones
EBP & W CTRL	=	Algorithme de rétropropagation améliorée ( <i>enhanced backpropagation</i> ) avec contrôle des poids
état du filtre	=	le vecteur $[u(n), u(n-1), \dots, u(n-N+1)]$ où $u(n)$ est l'entrée du filtre et $N$ est l'ordre du filtre
état du réseau	=	l'ensemble des vecteurs $\mathbf{X}_0$ et $\mathbf{X}_1$ du réseau
$\mu$	=	pas d'adaptation normalisé pour le filtre FIR
$\mu_1, \mu_2$	=	pas d'adaptation du réseau
NLMS	=	algorithme LMS normalisé
RNA FF2	=	réseau de neurones artificiel <i>feedforward</i> de deux couches



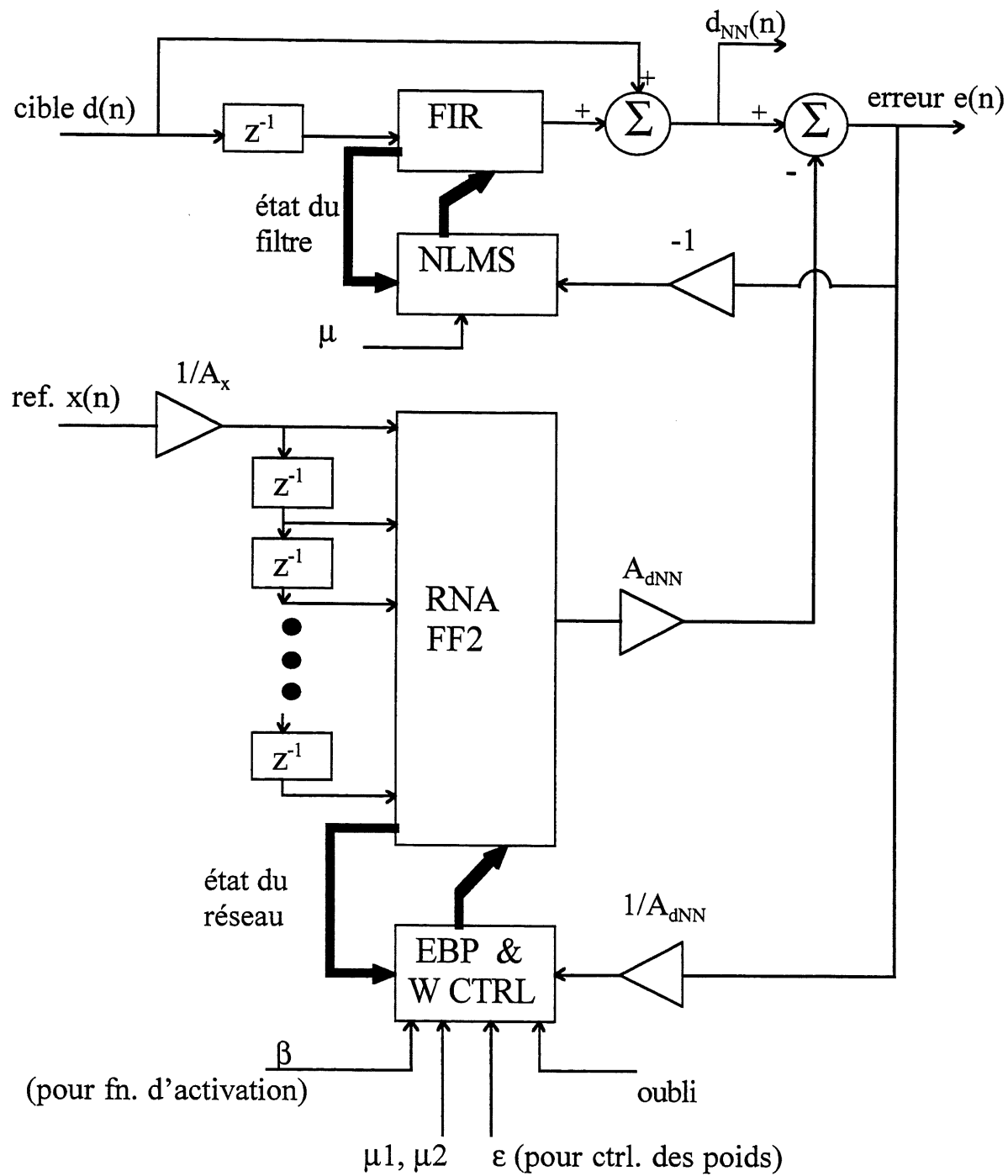


Figure 5.3 Structure pour l'identification basée sur le modèle I

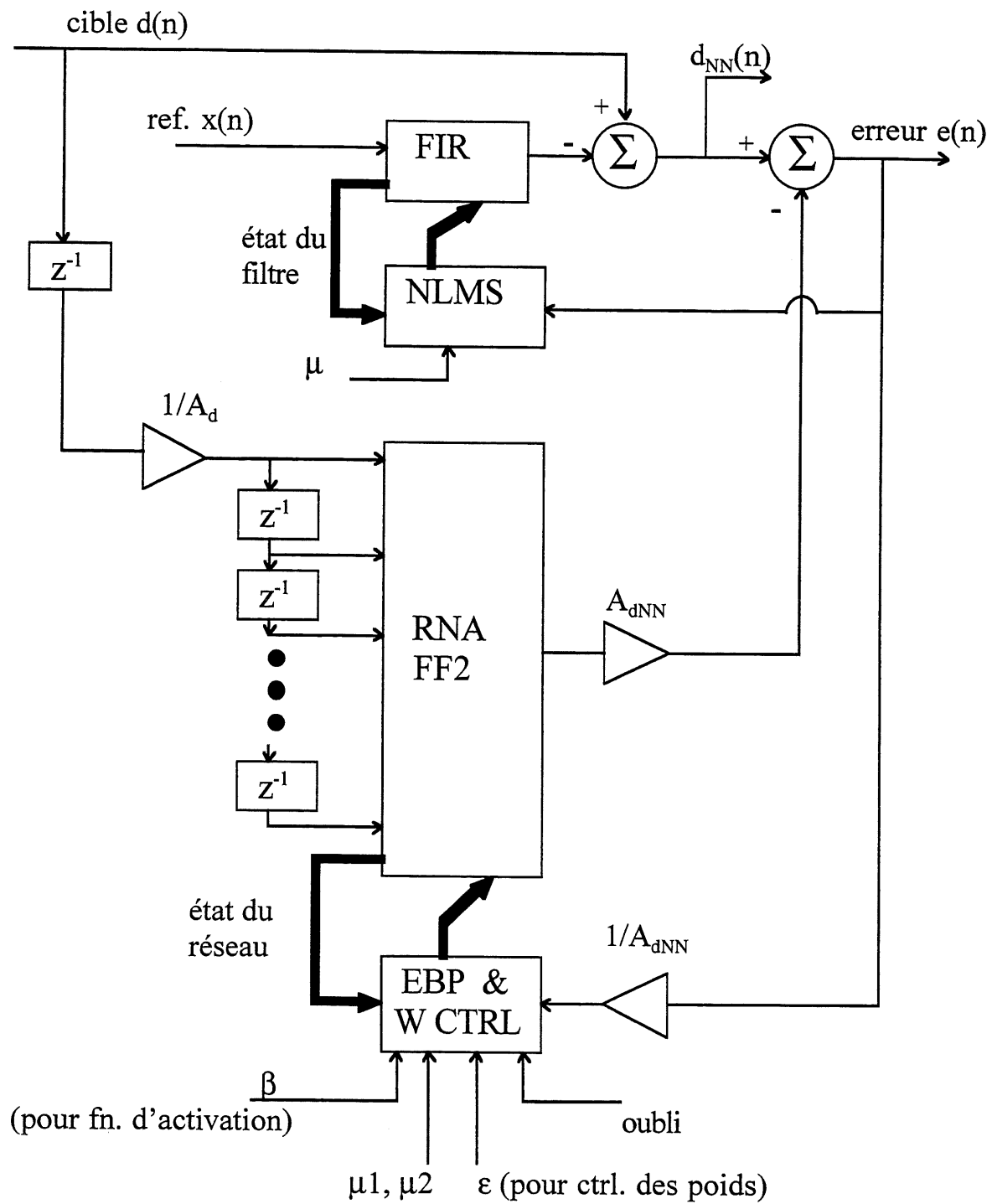


Figure 5.4 Structure pour l'identification basée sur le modèle II

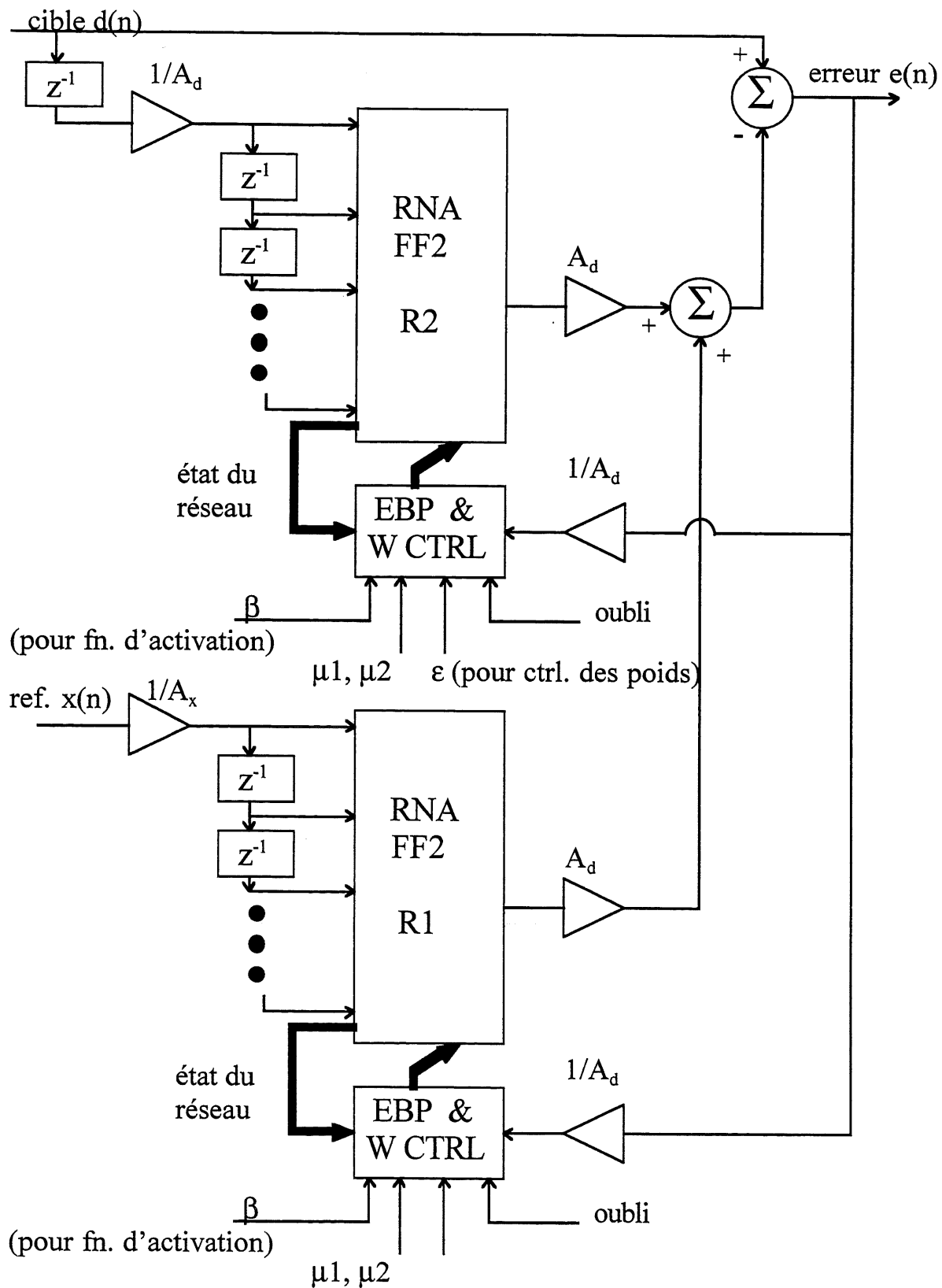


Figure 5.5 Structure pour l'identification basée sur le modèle III

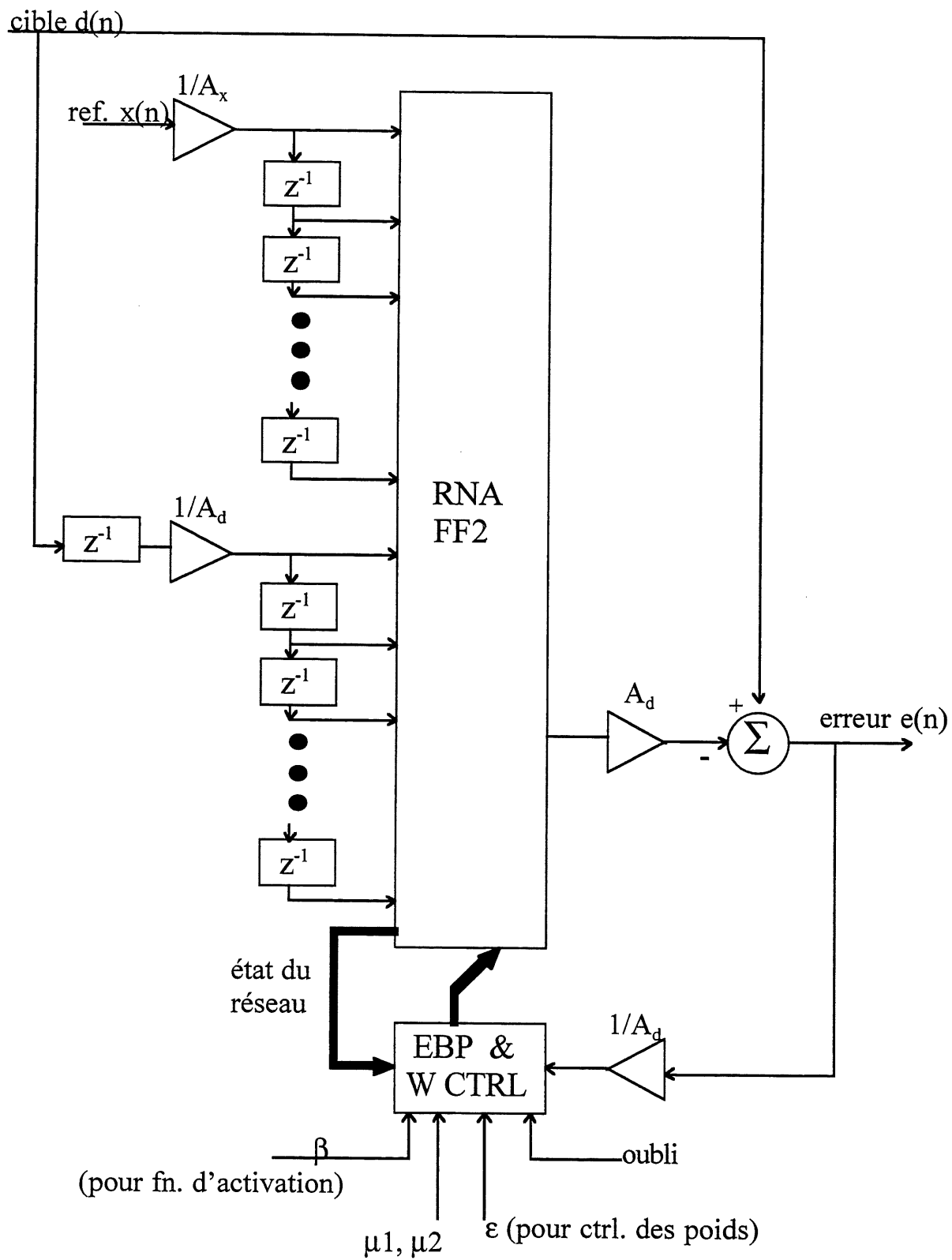


Figure 5.6 Structure pour l'identification basée sur le modèle IV

## 5.7 Méthode de vérification des modèles identifiés

Une fois que l'identification est faite, c'est-à-dire qu'on a atteint une convergence optimale, on fige les paramètres du modèle et on procède à une vérification. Pour cela, on excite le système physique et le modèle par le même signal.

Dans un premier temps, on vérifie le modèle pour des excitations en bruit blanc. On calcule l'erreur entre la sortie du système physique et celle du modèle. On peut aussi calculer le spectre d'énergie de la sortie du modèle et le comparer avec celui du système. Dans un deuxième temps, on vérifie les modèles pour des excitations en régime harmonique. Le but est de montrer que bien que l'apprentissage soit fait en large bande (excitation en bruit blanc), les modèles sont capables de représenter la non-linéarité du système, même pour des excitations en régime harmonique. Des comparaisons des spectres de sorties seront très pertinentes à cette fin.

Rappelons que pour faire l'identification des modèles récurrents, on utilise le signal cible,  $d(n)$ , comme référence (pour le filtre ou le réseau) au lieu de boucler la sortie du modèle à l'entrée (figures 5.3 à 5.6). Cela nous permet d'éviter des problèmes d'instabilité, de minimum local et de convergence trop lente. Cependant, pour la vérification de chaque modèle, on doit retourner la sortie du modèle même à l'entrée, comme illustrée dans la figure 5.7, pour la vérification du modèle III par exemple.

Lors de la vérification d'un modèle en mode récurrent, on doit s'attendre à une dérive de la sortie du modèle par rapport à la sortie du système. Cette dérive est due au fait que les paramètres du modèle sont identifiés avec une certaine

marge d'erreur. L'erreur sur chaque paramètre contribue à une erreur à la sortie du modèle. En bouclant la sortie du modèle à l'entrée, cette erreur à la sortie est retournée à l'entrée, ce qui provoque la dérive de la sortie du modèle avec le temps.

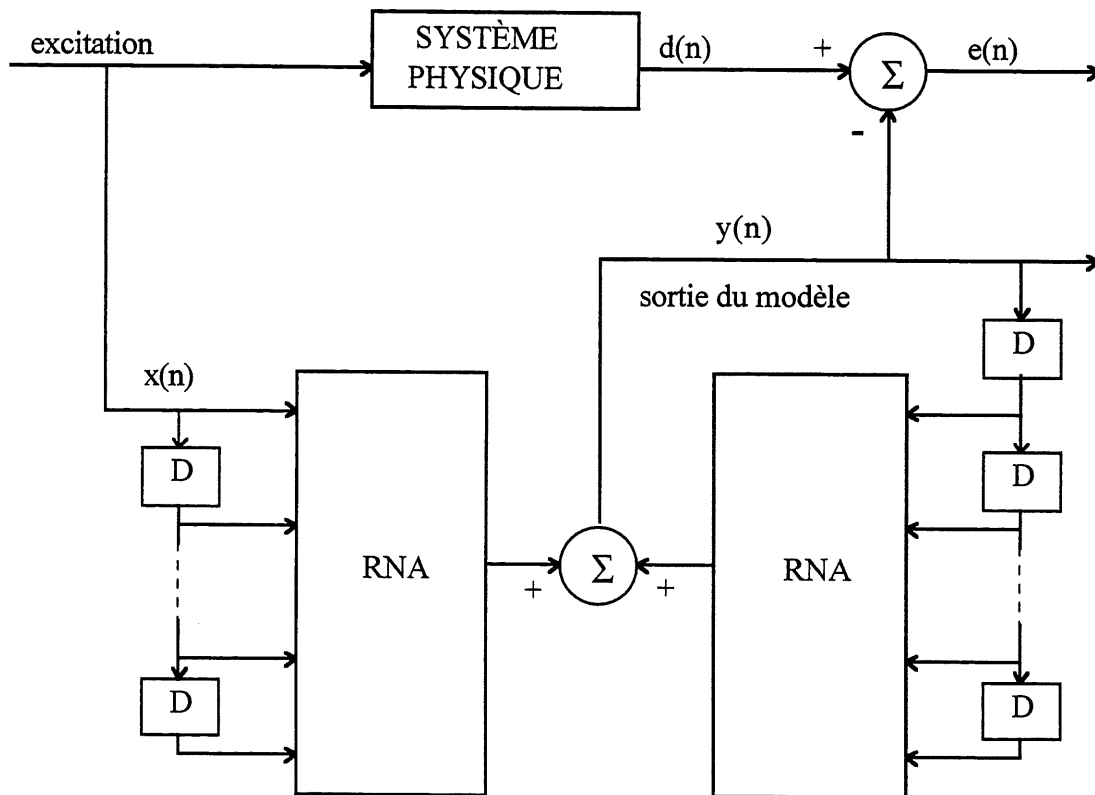


Figure 5.7 Vérification du modèle III en mode récursif

## 6. RÉSULTATS DES IDENTIFICATIONS NON-LINÉAIRES

### 6.1 Introduction

On présente dans ce chapitre les résultats des identifications basées sur les théories du chapitre précédent. On commence par présenter les résultats des identifications basées sur chacun des quatre modèles. On procède aussi à une vérification de la validité des modèles retenus. Ensuite, on considère une optimisation du modèle IV. Finalement, on commente sur les résultats obtenus. Toutes les programmations sont faites avec le logiciel LabVIEW. La fréquence d'échantillonnage est à 120.0 Hz dans toutes les identifications.

### 6.2 Identification basée sur le modèle I

Vu que le modèle I est une extension directe du modèle ARMA linéaire, l'identification basée sur ce premier est faite de la manière suivante. On commence par faire une identification du système basée sur le modèle ARMA linéaire comme décrite dans la section 4.5. Une fois qu'on a atteint une convergence optimale avec ce modèle, on arrête l'adaptation et on conserve les coefficients  $\{a_1, a_2\}$  de la partie autorégressive. On procède ensuite à une identification basée sur le modèle I comme illustrée dans la figure 5.3.

Le filtre FIR dans cette structure est initialisé à  $\{a_1, a_2\}$  et il n'est pas adapté dans la première phase de l'identification. La ligne à retard qui sert d'entrée au réseau contient 127 éléments de retard. Le réseau a 128 points d'entrées et 64 neurones. Les poids du réseau sont initialisés comme décrits dans la section 5.4.

On commence l'adaptation du réseau avec les paramètres suivants: oubli = 0.995,  $\varepsilon = 0.1$ , et  $\mu_1 = \mu_2 = -10$  dB. La convergence est assez rapide: on atteint

-15 dB après environ 1300 itérations et -30 dB après environ 4000 itérations. À ce point, la convergence est beaucoup plus lente. Après 20,000 itérations, la convergence varie entre -31 et -37 dB. Là, on laisse varier les coefficients du filtre FIR avec un pas  $\mu_{\text{norm}} = -40$  dB et les paramètres du réseau sont ajustés: oubli = 0.999,  $\varepsilon = 0.1$ , et  $\mu_1 = \mu_2 = -20$  dB. Après 45,000 itérations, la convergence varie entre -32 et -37 dB. On arrête l'adaptation à ce point.

Un bloc du signal cible est montré dans la figure 6.1 et l'erreur d'identification, à la fin de l'adaptation, est montrée dans la figure 6.2. Les spectres de ces signaux sont montrés dans la figure 6.3.

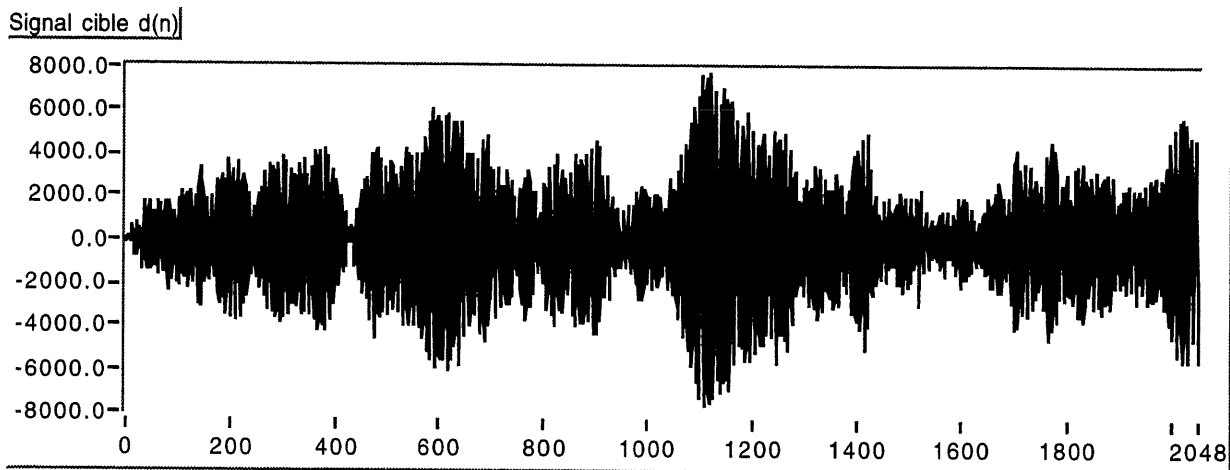


Figure 6.1 Un bloc du signal cible

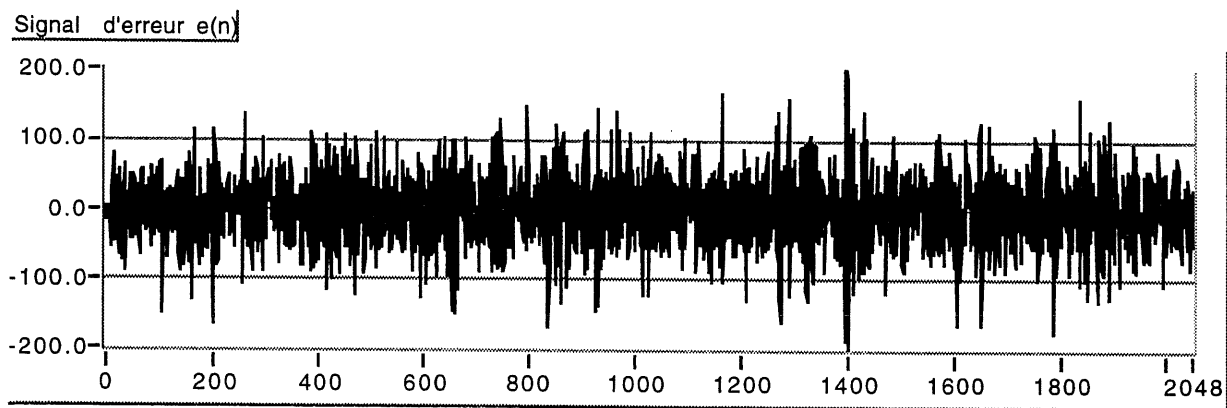


Figure 6.2 Erreur à la fin de l'identification (modèle I)



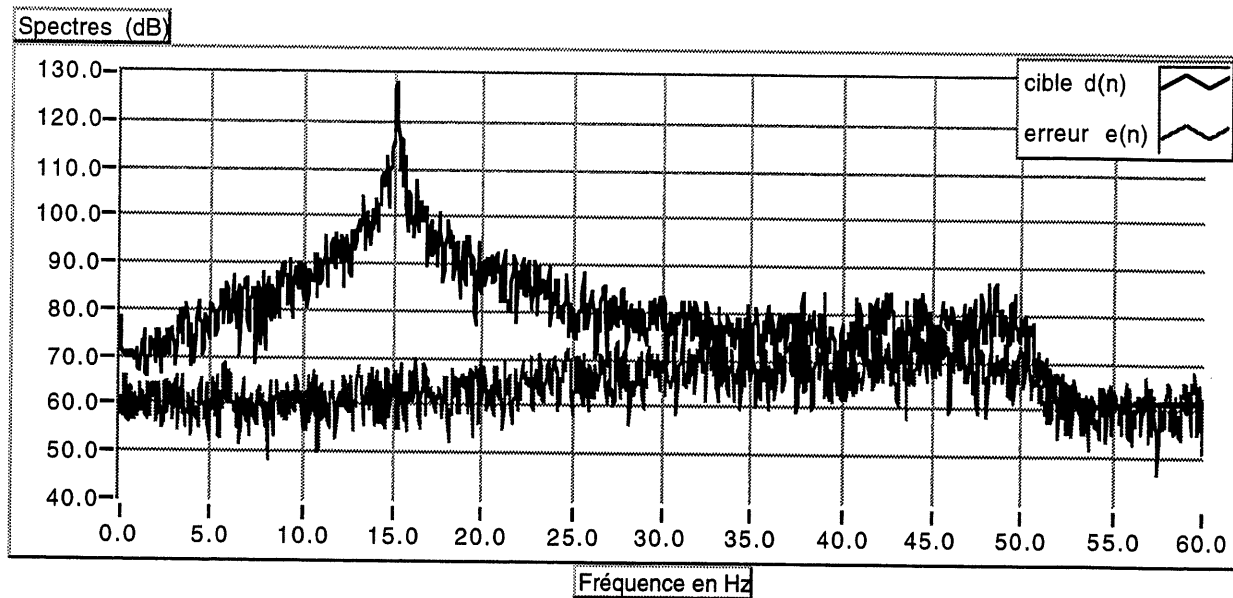


Figure 6.3 Spectres des signaux de cible et d'erreur (modèle I)

### 6.3 Identification basée sur le modèle II

Similairement à l'identification basée sur le modèle I, l'identification par le modèle II est faite de la manière suivante. On commence par une identification du système basée sur le modèle ARMA linéaire comme décrite dans la section 4.5, en utilisant 160 coefficients pour la partie MA. Une fois qu'on a atteint une convergence optimale avec ce modèle, on arrête l'adaptation et on conserve les coefficients  $\{b_0, b_1, \dots, b_{N-1}\}$  de la partie MA. On procède ensuite à une identification basée sur le modèle II comme illustrée dans la figure 5.4.

Le filtre FIR dans cette structure est initialisé à  $\{b_0, b_1, \dots, b_{N-1}\}$  et il n'est pas adapté dans la première phase de l'identification. La ligne à retard qui sert d'entrée au réseau contient 7 éléments de retard. Le réseau a 8 points d'entrées et 50 neurones. Les poids du réseau sont initialisés comme décrits dans la section 5.4.

On commence l'adaptation du réseau avec les paramètres suivants: oubli = 0.995,  $\varepsilon = 0.1$ , et  $\mu_1 = \mu_2 = -10$  dB. La convergence est très rapide au début. Une fois qu'on a atteint une convergence de -30 dB, on ajuste les paramètres d'adaptation : oubli = 0.999,  $\varepsilon = 0.1$ ,  $\mu_1 = \mu_2 = -16$  dB. Après 20,000 itérations, on laisse varier les coefficients  $\{b_0, b_1, \dots, b_{N-1}\}$  avec un pas  $\mu_{\text{norm}} = -30$  dB et on place  $\mu_1 = \mu_2 = -20$  dB. Après 75,000 itérations, la convergence varie entre -33 et -39 dB. On arrête l'adaptation ici. L'erreur d'identification pour le dernier bloc de signal est donnée dans la figure 6.4. Les spectres de la cible et de l'erreur sont donnés dans la figure 6.5.

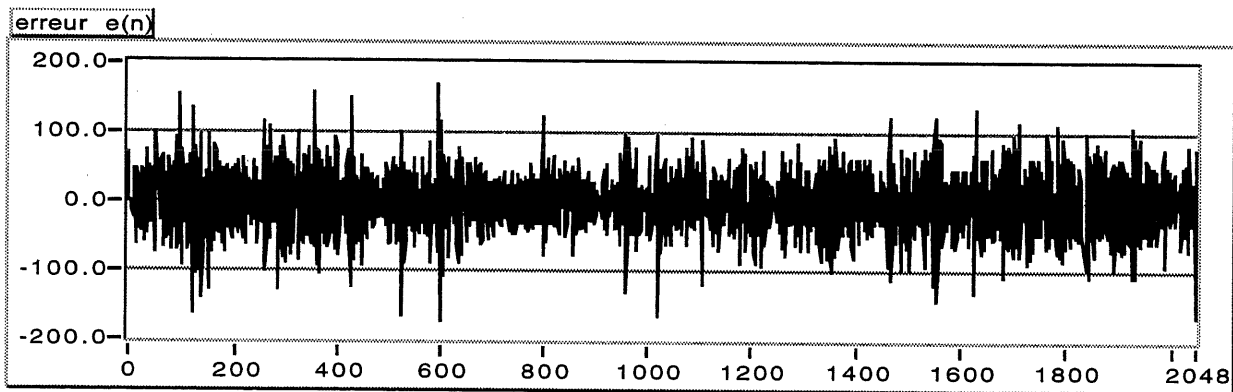


Figure 6.4 Erreur à la fin de l'identification (modèle II)

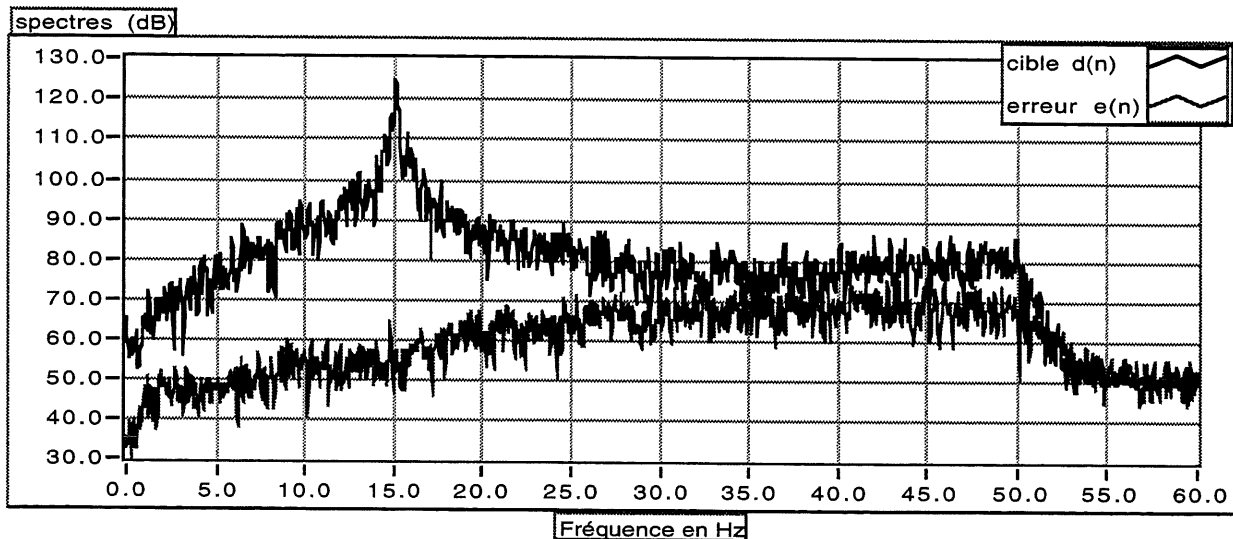


Figure 6.5 Spectres des signaux de cible et d'erreur (modèle II)

#### 6.4 Identification basée sur le modèle III

Cette identification est faite d'après la structure de la figure 5.5. La ligne à retard pour le réseau R1 contient 127 éléments de retard et celle pour le réseau R2 contient 7 éléments de retard. R1 contient 128 entrées et 48 neurones. R2 contient 8 entrées et 48 neurones. Les initialisations sont faites indépendamment.

On commence l'identification avec les paramètres suivants, pour chaque réseau : oubli = 0.995,  $\varepsilon = 0.1$ ,  $\mu_1 = \mu_2 = -10$  dB. La convergence est très rapide au début. On atteint -15 dB après 100 itérations, -20 dB après 800 itérations et -25 dB après 1400 itérations. Après 5000 itérations, la convergence est environ -32 dB et après 20,000 itérations elle varie entre -36 et -41 dB. À ce point on ajuste les paramètres d'adaptation: oubli = 0.999,  $\varepsilon = 0.1$ ,  $\mu_1 = \mu_2 = -20$  dB. Après 30,000 itérations, la convergence varie entre -37 et -42 dB. On n'observe aucune amélioration même après 45,000 itérations. On arrête l'adaptation là.

Le signal d'erreur à la fin de l'adaptation est montré dans la figure 6.6. Les spectres des signaux de cible et d'erreur sont donnés dans la figure 6.7.

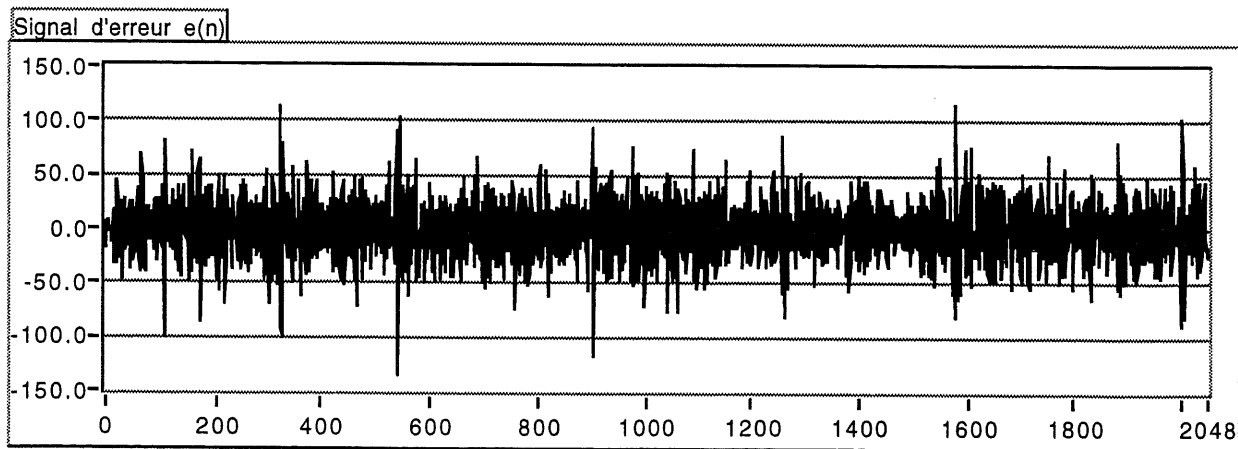


Figure 6.6 Erreur après l'adaptation du modèle III

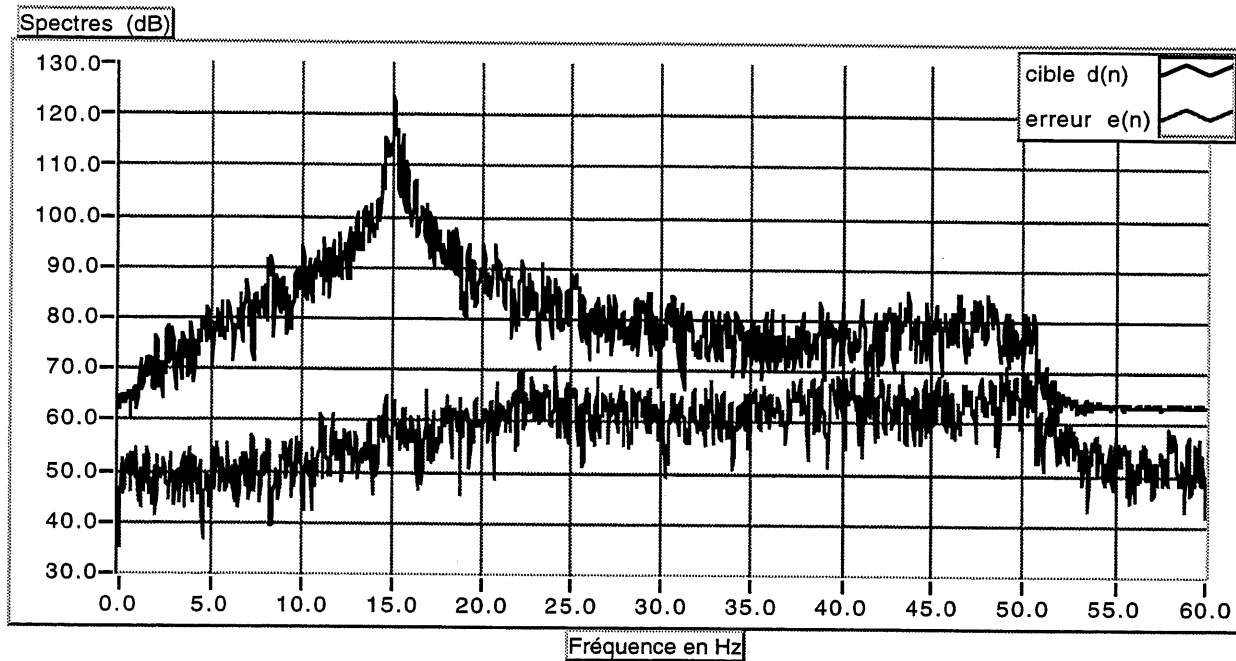


Figure 6.7 Spectres des signaux de cible et d'erreur (modèle III)

## 6.5 Identification basée sur le modèle IV

L'identification est faite d'après la structure de la figure 5.6. La ligne à retard avec entrée  $x(n)$  contient 127 éléments de retard et l'autre contient 7 éléments. Le réseau contient 136 entrées et 48 neurones.

On commence l'adaptation avec les paramètres suivants: oubli = 0.995,  $\varepsilon = 0.1$ ,  $\mu_1 = \mu_2 = -10$  dB. La convergence est très lente au début. Après environ 6000 itérations, l'erreur baisse plus rapidement. Après 20,000 itérations, la convergence varie entre -37 et -42 dB et on ajuste les paramètres d'adaptation: oubli = 0.999,  $\varepsilon = 0.1$ ,  $\mu_1 = \mu_2 = -20$  dB. Après 45,000 itérations la convergence varie entre -38 et -42 dB. L'adaptation est arrêtée ici. L'erreur d'identification pour le dernier bloc de signal est donnée dans la figure 6.8. Les spectres des signaux de cible et d'erreur sont donnés dans la figure 6.9.

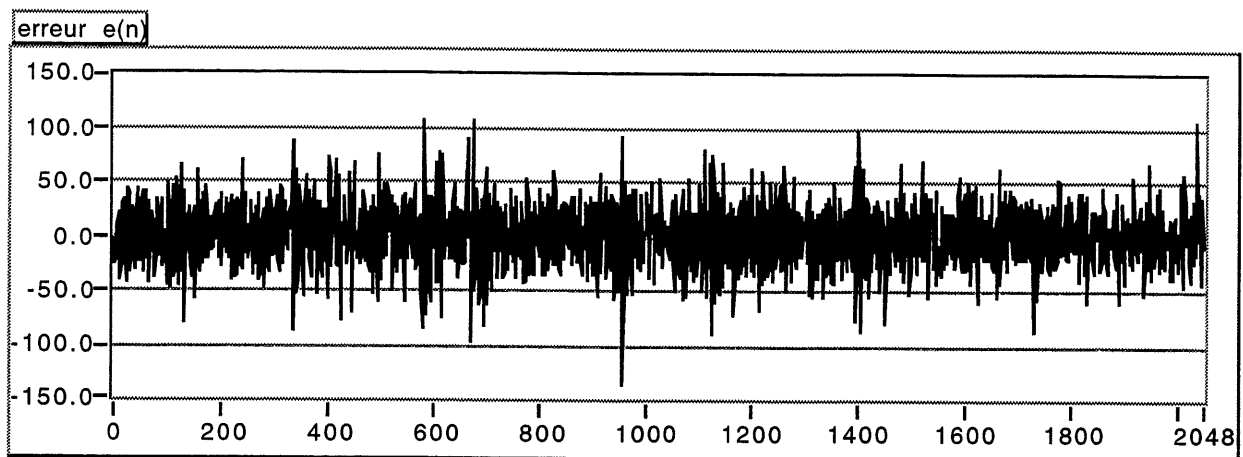


Figure 6.8 Erreur d'identification pour le dernier bloc de signal (modèle IV)

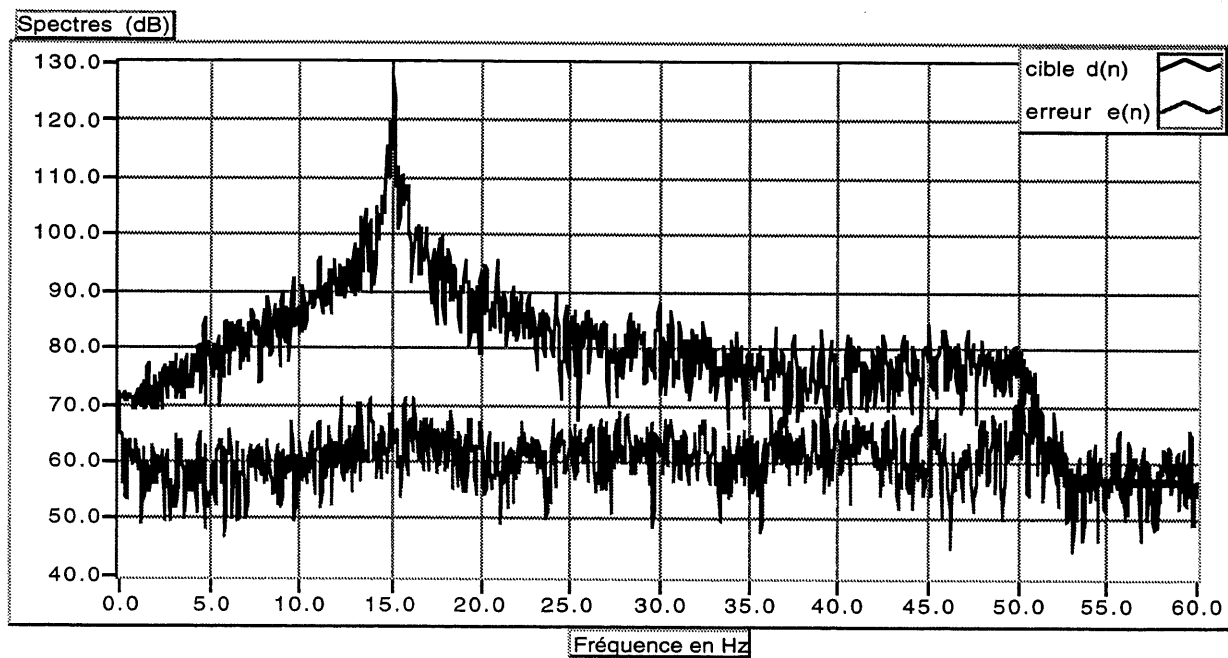


Figure 6.9 Spectres des signaux de cible et d'erreur pour modèle IV

## 6.6 Vérification des modèles identifiés pour excitation en bruit blanc

La vérification des modèles identifiés, pour des excitations en bruit blanc, est faite comme décrite dans la section 5.7. On ne présente ici, que les résultats pour les modèles III et IV seulement, car les modèles I et II ne sont pas satisfaisants en vue de leur convergence relativement faible. La figure 6.10(a) nous montre sur le même graphique les réponses du système physique et du modèle III pour la même excitation (en bruit blanc). La figure 6.10(b) nous montre la différence entre ces deux réponses, c'est-à-dire l'erreur de modélisation. La figure 6.10(c) nous montre les spectres des réponses du système et du modèle, ainsi que le spectre de l'erreur. Les résultats de la vérification du modèle IV, pour les excitations en bruit blanc, sont donnés dans les figures 6.11(a) à 6.11(c).

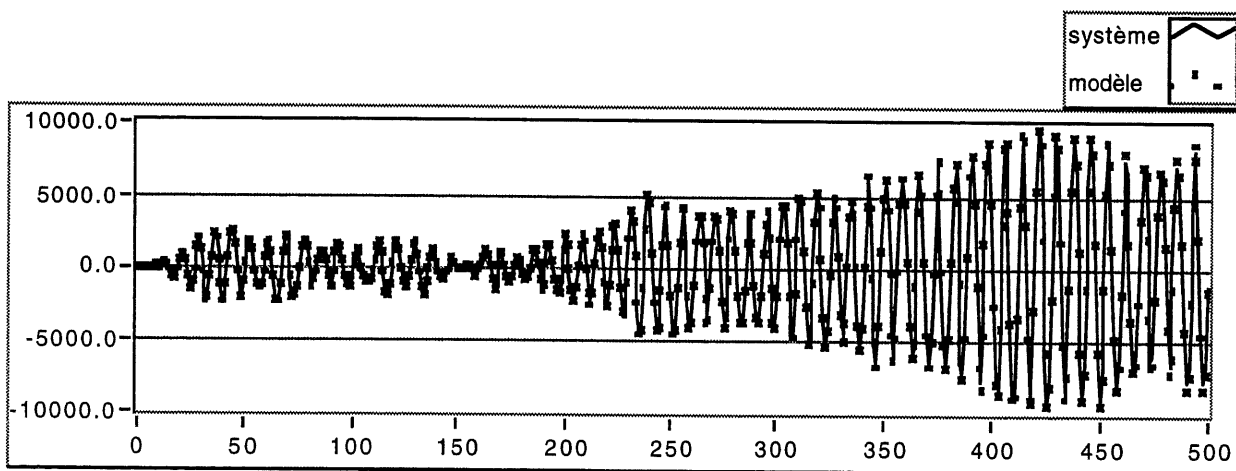


Figure 6.10(a) Réponse du modèle III en comparaison avec celle du système

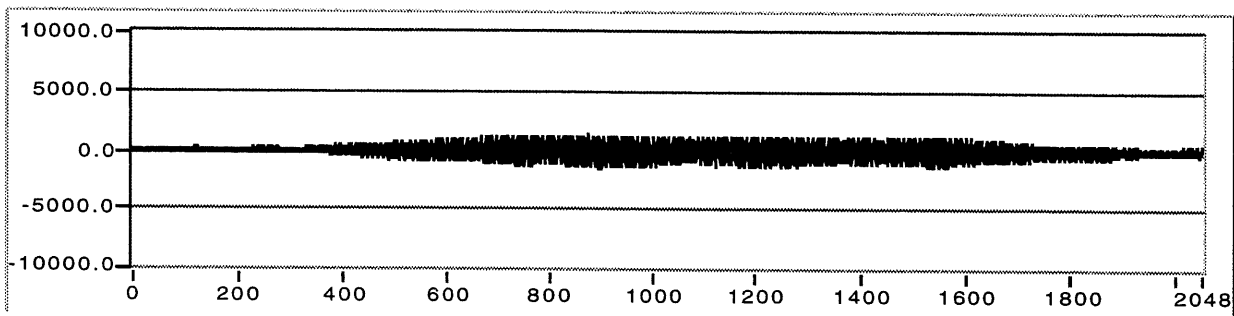


Figure 6.10(b) Erreur de modélisation (modèle III)

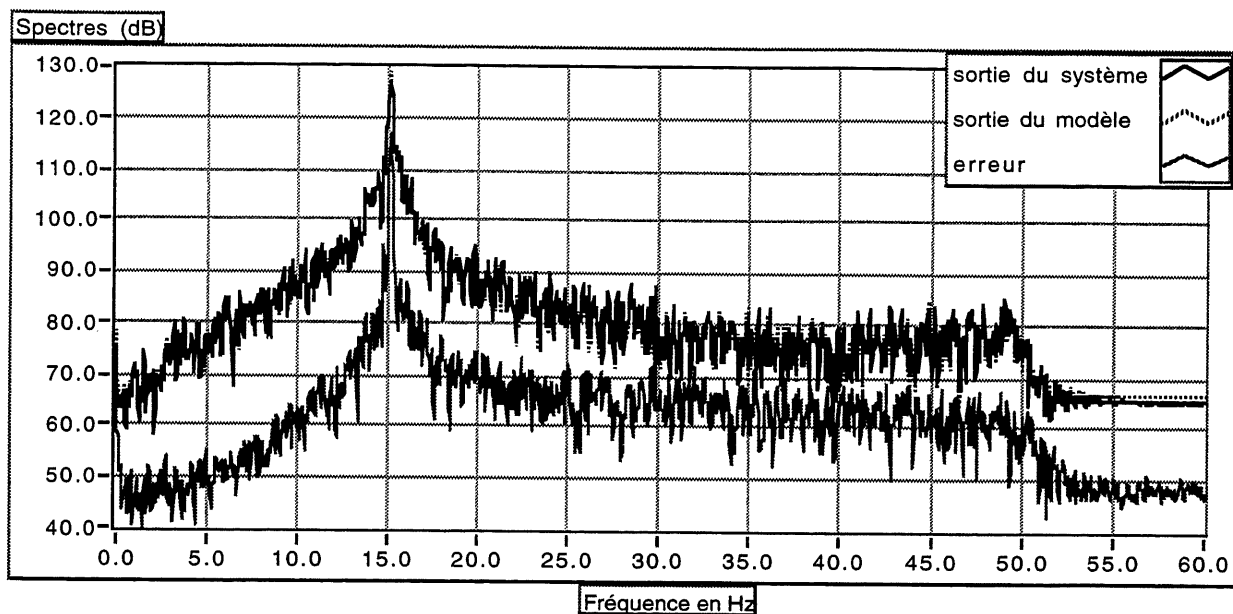


Figure 6.10(c) Spectres des réponses du système et du modèle III

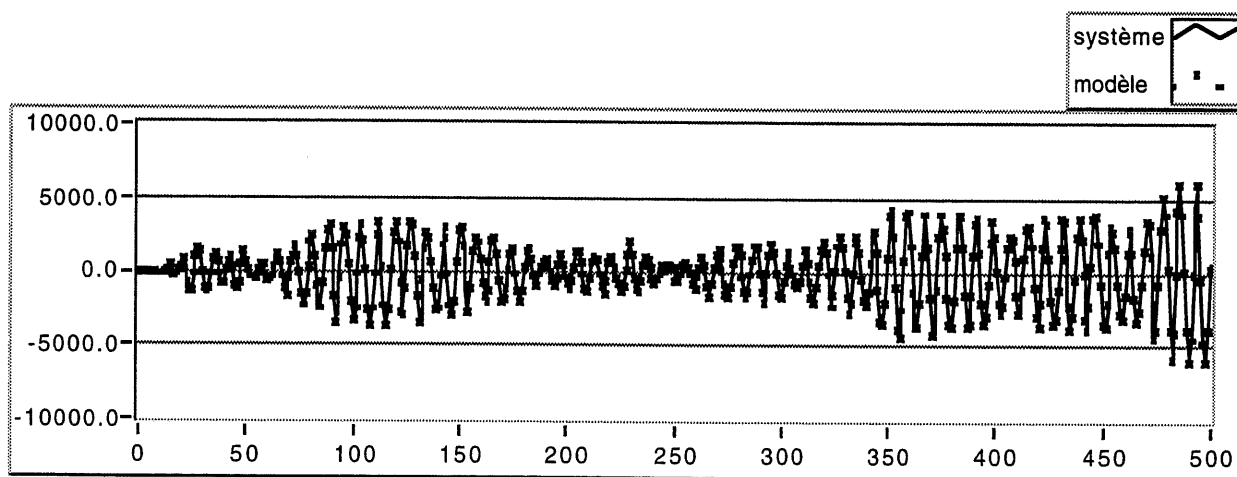


Figure 6.11(a) Réponse du modèle IV en comparaison avec celle du système

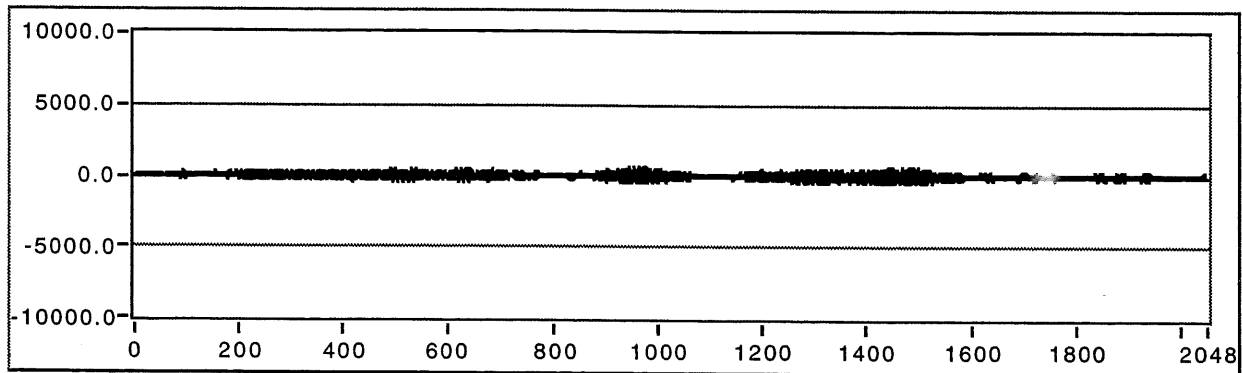


Figure 6.11(b) Erreur de modélisation (modèle IV)

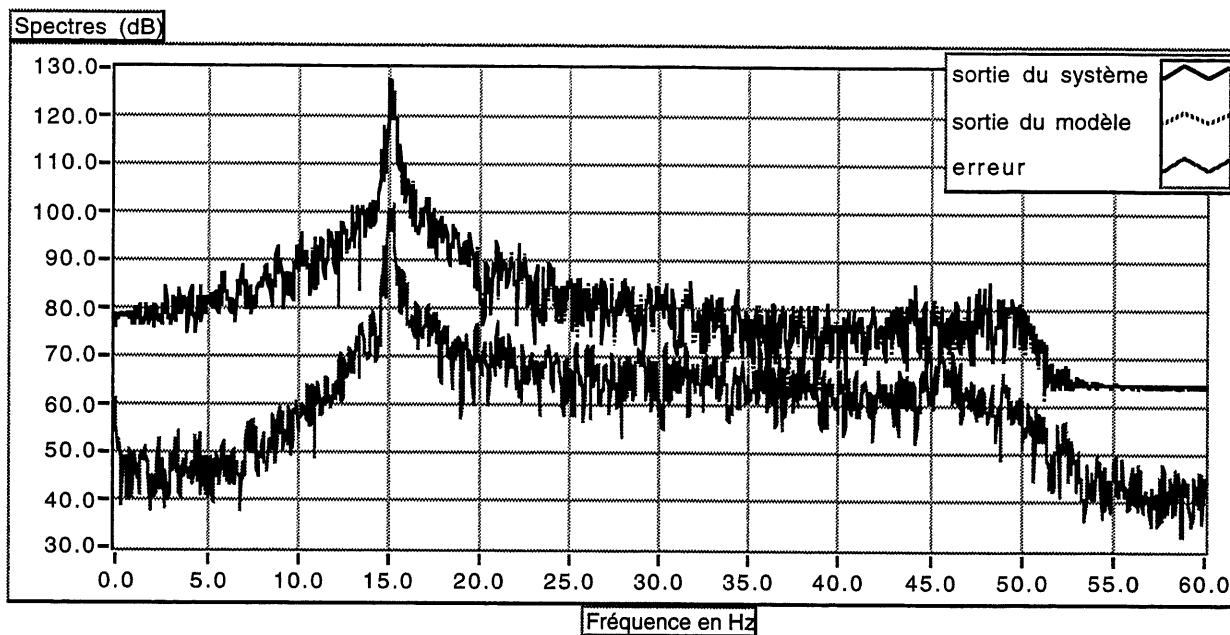


Figure 6.11(c) Spectres des réponses du système et du modèle IV

En observant la figure 6.10(b), on remarque une erreur assez importante pour le modèle III. Il y a deux sources qui contribuent à cette erreur. Premièrement, la sortie du système est contaminée par un bruit non-corrélé à l'entrée. L'énergie du bruit est concentrée dans la proximité de 15 Hz (fréquence de résonance). Deuxièmement, l'erreur sur chaque paramètre identifié contribue à une erreur à la sortie du modèle. Puisque les modèles sont vérifiés en mode récursif (voir la figure 5.7), cette erreur à la sortie est retournée à l'entrée, ce qui entraîne une dérive de la sortie du modèle par rapport à la sortie du système. Cette dérive n'est pas très grave pour les premiers 300 échantillons, mais elle est bien plus importante par la suite (figure 6.10(b)).

En examinant la figure 6.11(b), on remarque que l'erreur pour le modèle IV est moins importante que pour le modèle III. Cela s'explique par le fait que le modèle IV, qui est le plus général, tient compte de toutes les corrélations entre les signaux  $x(n)$  et  $d(n)$ , donc la dérive de la sortie est moins importante.



## 6.7 Vérification des modèles pour excitation en régime harmonique

On présente dans cette section les résultats des vérifications des modèles III et IV, pour des excitations en régime harmonique. La méthode de vérification est similaire à celle pour la vérification en large bande (en bruit blanc), sauf que cette fois-ci, les excitations sont des sinusoïdes. Une comparaison des réponses temporelles n'est pas très intéressante à cause de la présence du bruit dans le système et de la dérive de la sortie du modèle (due à l'erreur sur chaque paramètre identifié) par rapport à la sortie du système. Cependant, une comparaison des réponses spectrales permet de faire ressortir les harmoniques dues à la non-linéarité. Ces spectres nous permettent aussi de comparer les modèles III et IV, en ce qui concerne la capacité de représenter la non-linéarité du système. Les résultats pour la vérification du modèle III, pour des excitations en régime harmonique, sont donnés dans les figures 6.12(a) à 6.12(f). Les résultats pour la vérification du modèle IV sont donnés dans les figures 6.13(a) à 6.13(f). Dans tous les cas, l'amplitude du signal d'excitation est 8000.

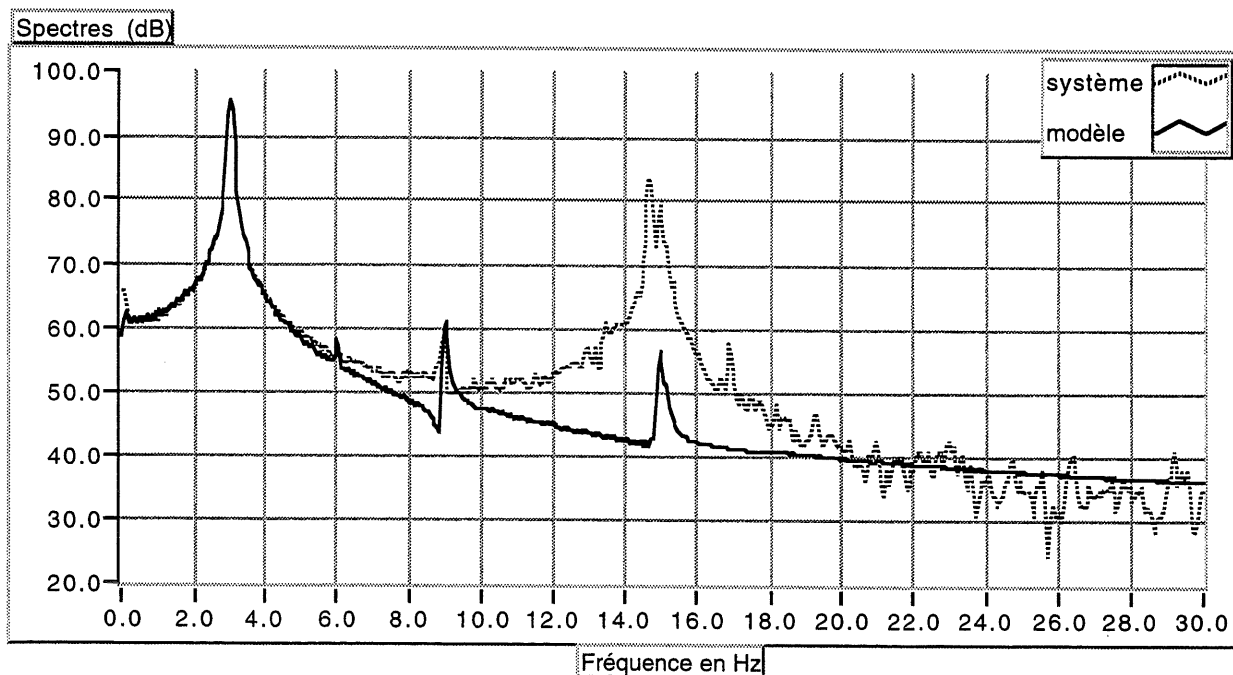


Figure 6.12(a) Vérification du modèle III: excitation à 3.0 Hz

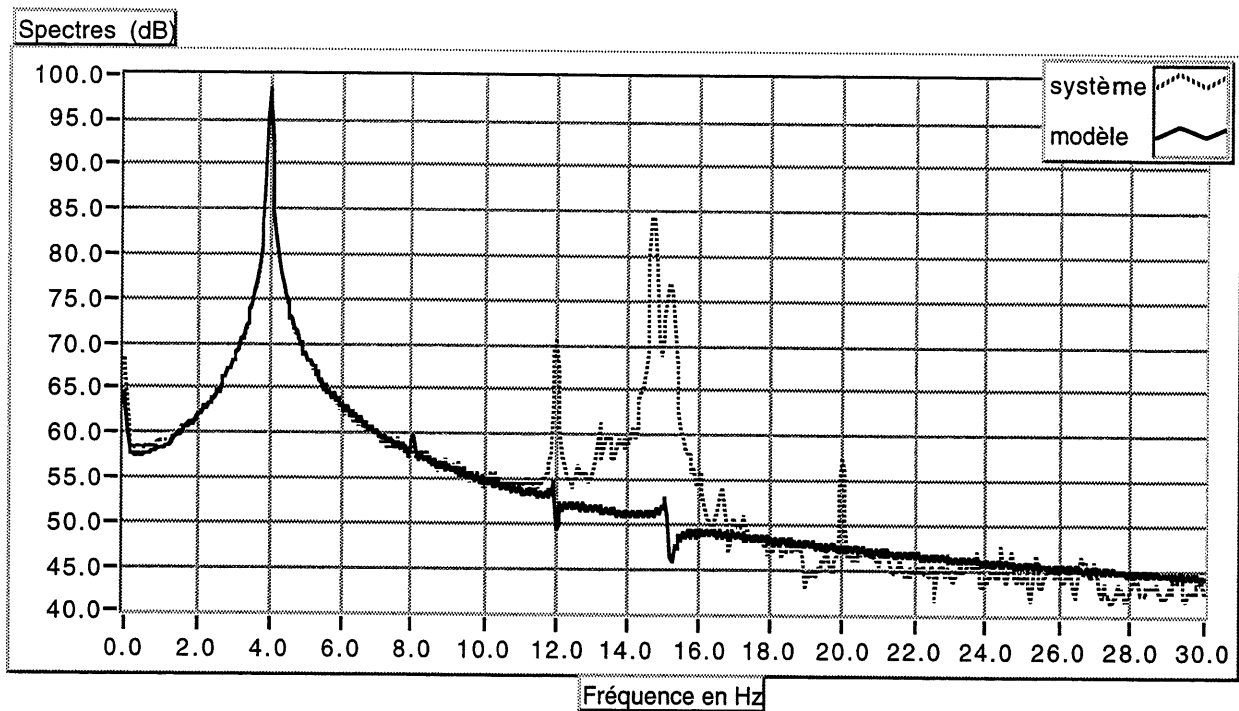


Figure 6.12(b) Vérification du modèle III: excitation à 4.0 Hz

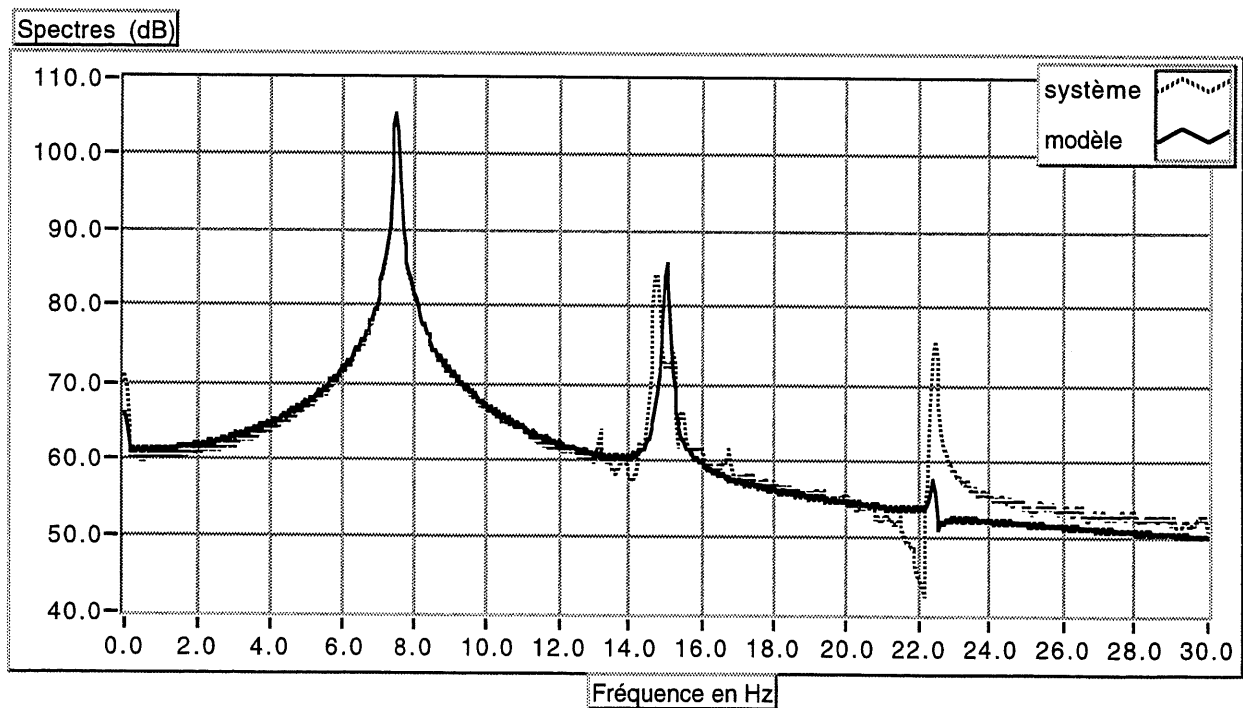


Figure 6.12(c) Vérification du modèle III: excitation à 7.5 Hz

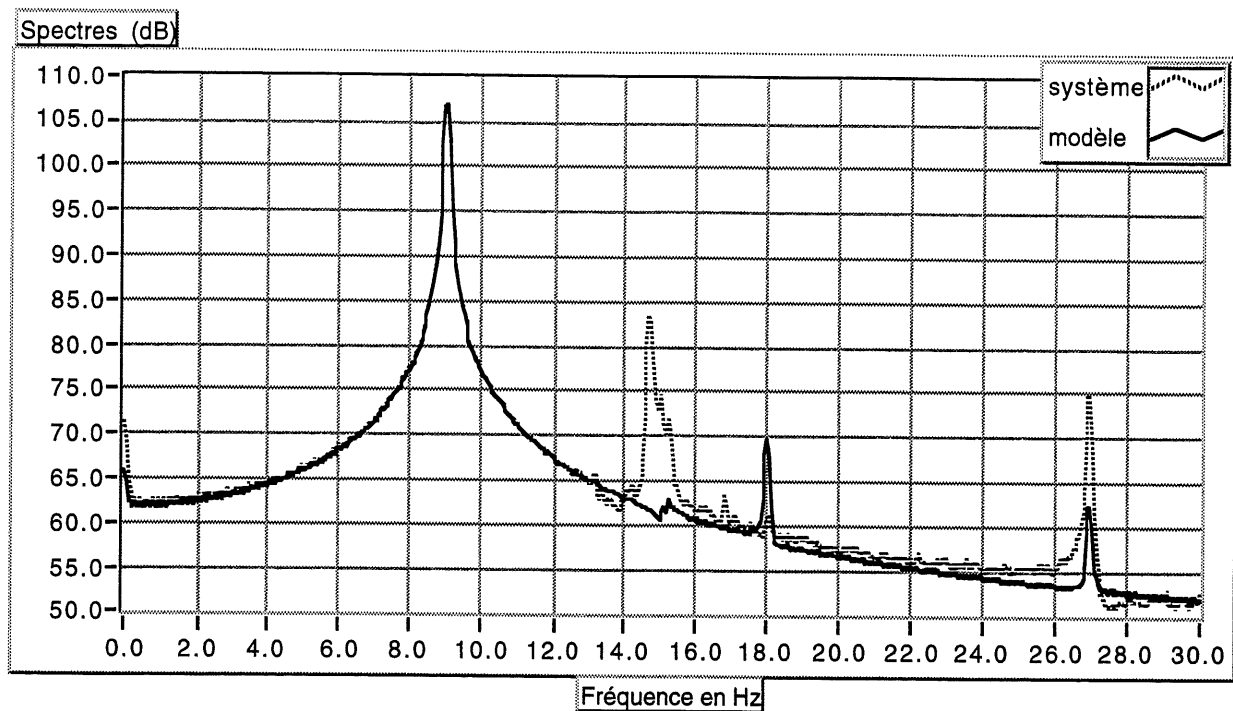


Figure 6.12(d) Vérification du modèle III: excitation à 9.0 Hz

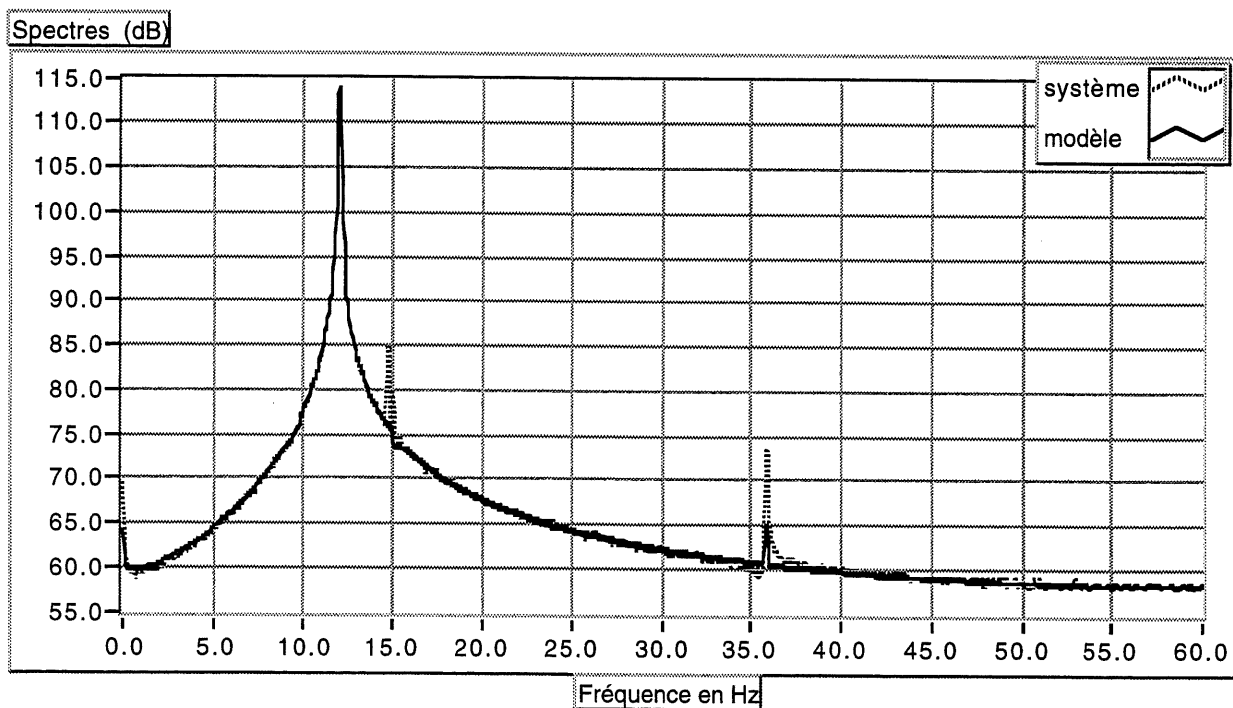


Figure 6.12(e) Vérification du modèle III: excitation à 12.0 Hz

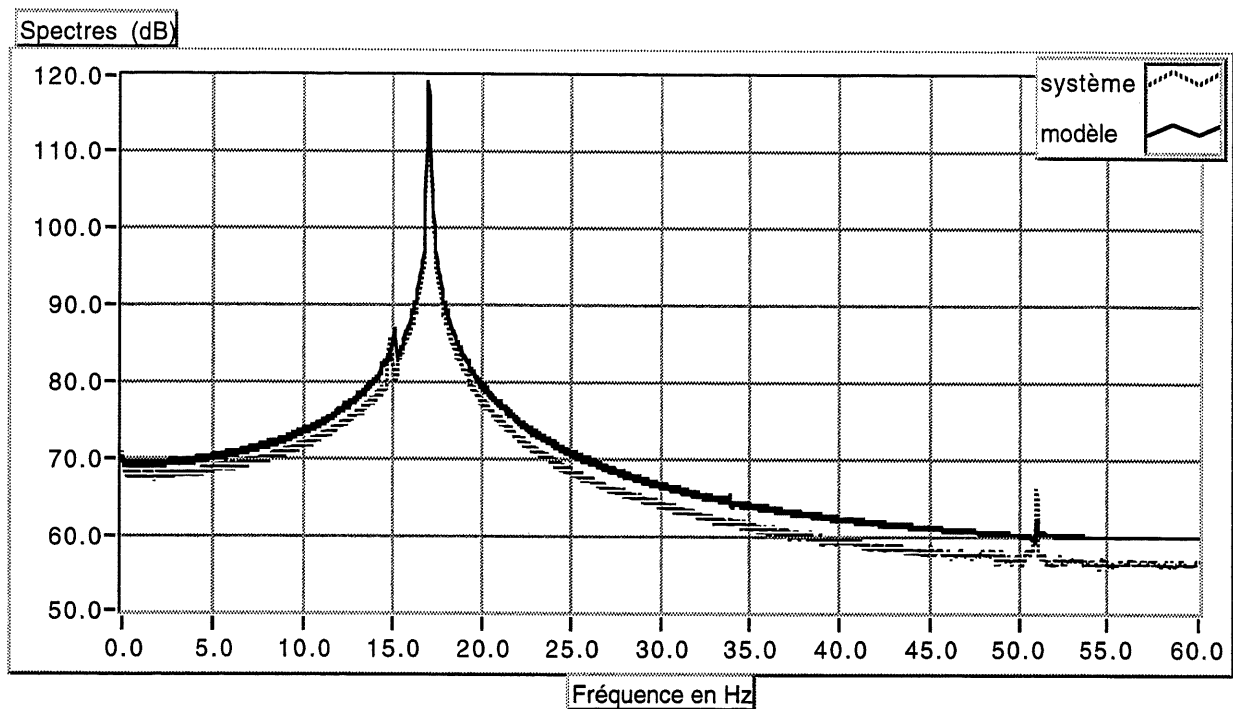


Figure 6.12(f) Vérification du modèle III: excitation à 17.0 Hz

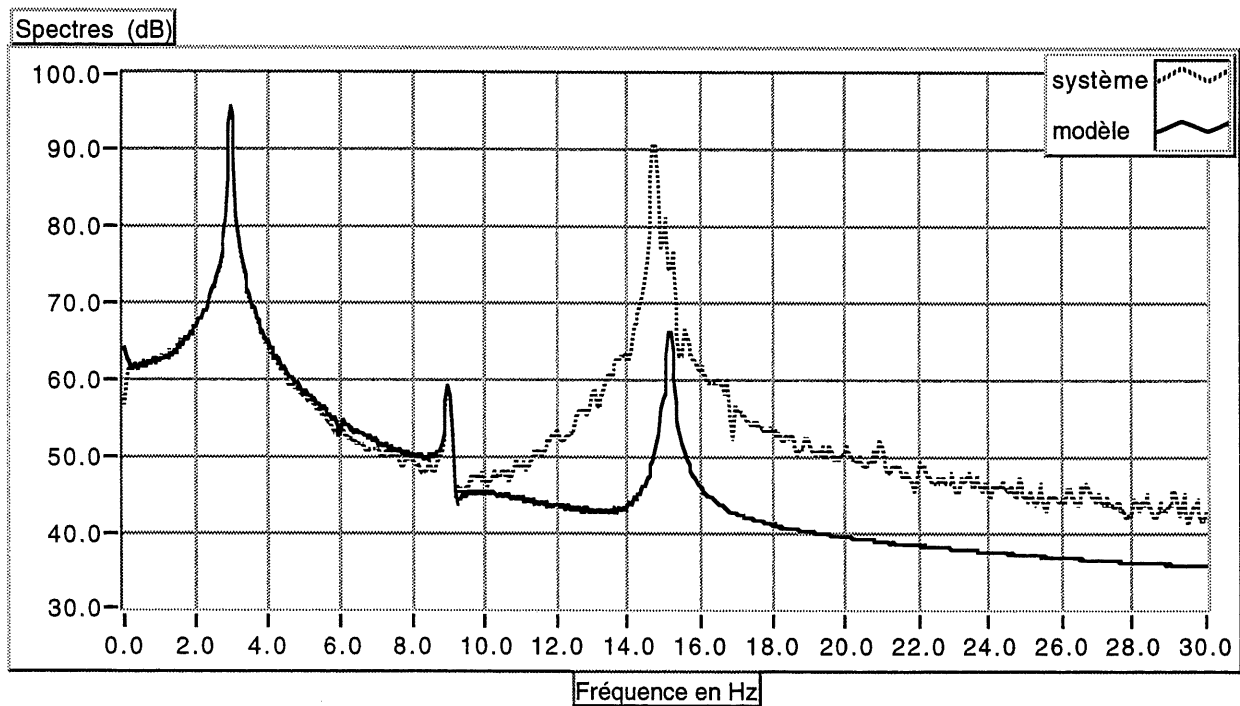


Figure 6.13(a) Vérification du modèle IV: excitation à 3.0 Hz

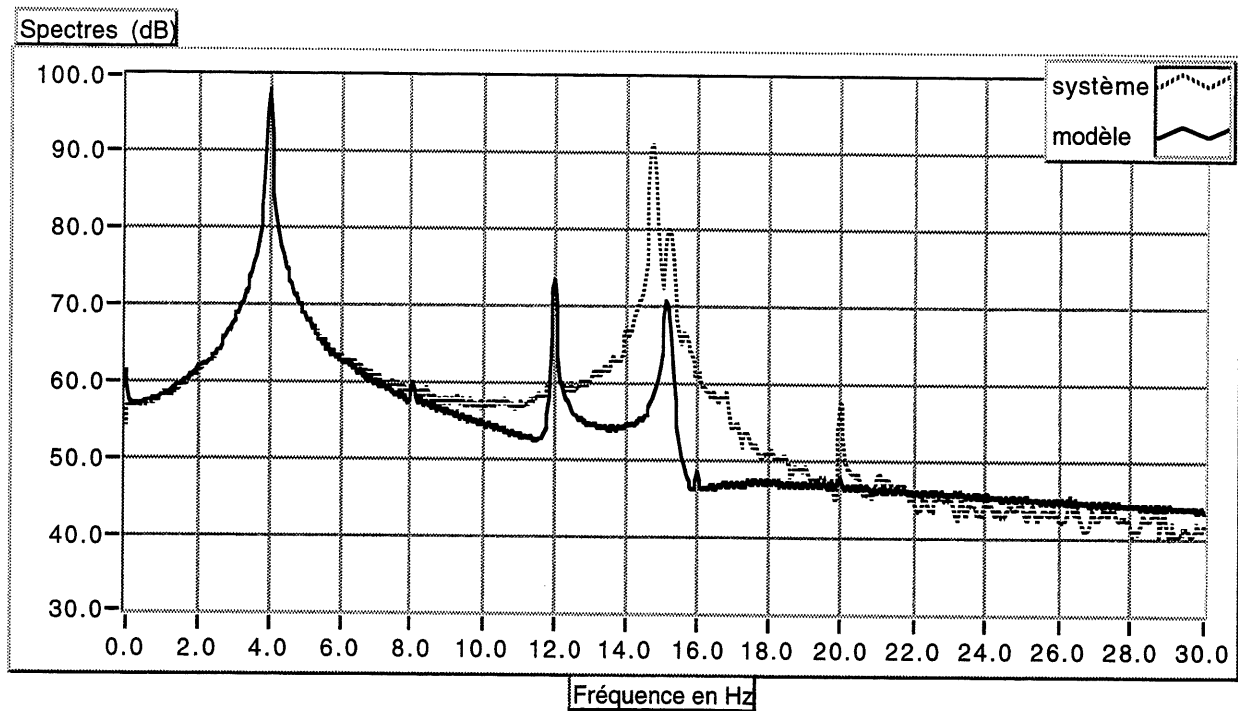


Figure 6.13(b) Vérification du modèle IV: excitation à 4.0 Hz

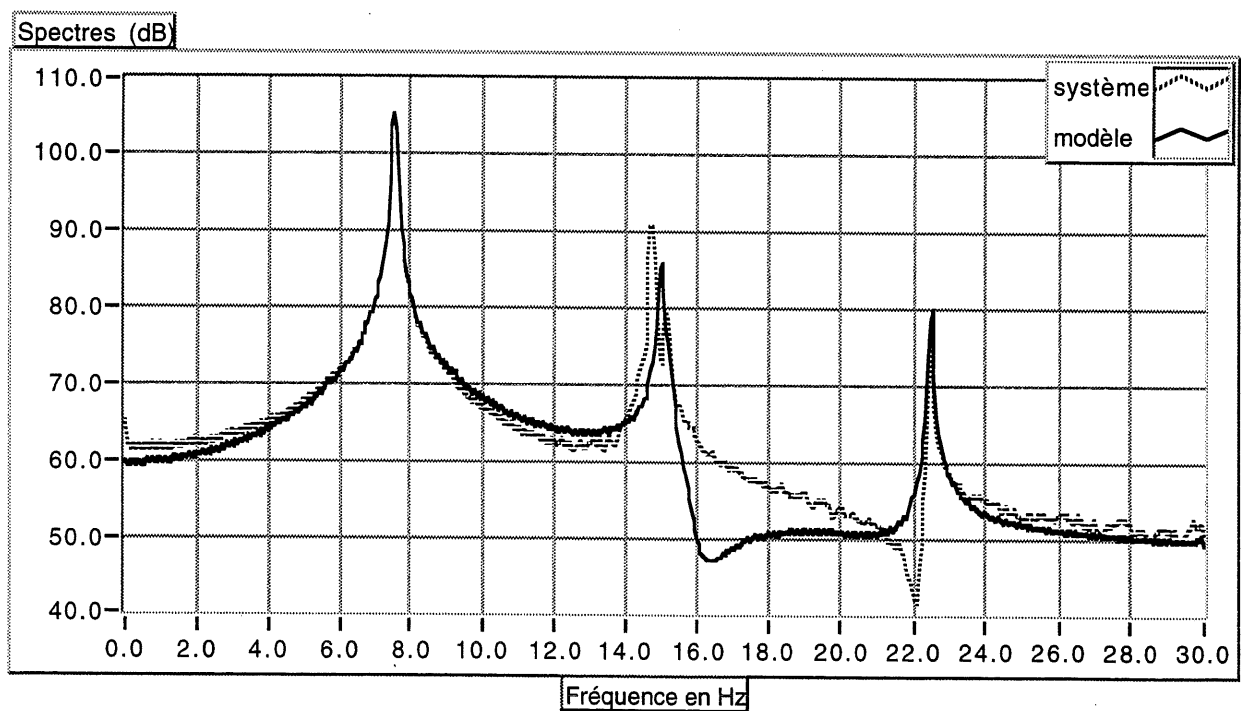


Figure 6.13(c) Vérification du modèle IV: excitation à 7.5 Hz

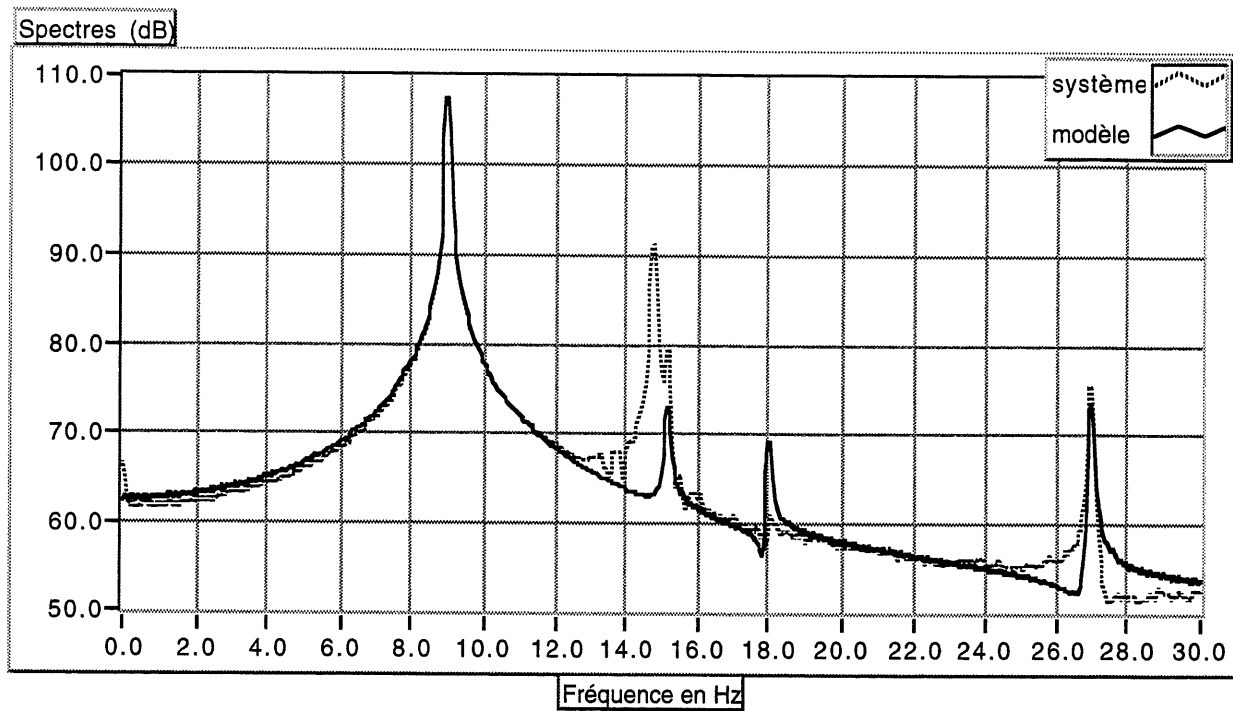


Figure 6.13(d) Vérification du modèle IV: excitation à 9.0 Hz

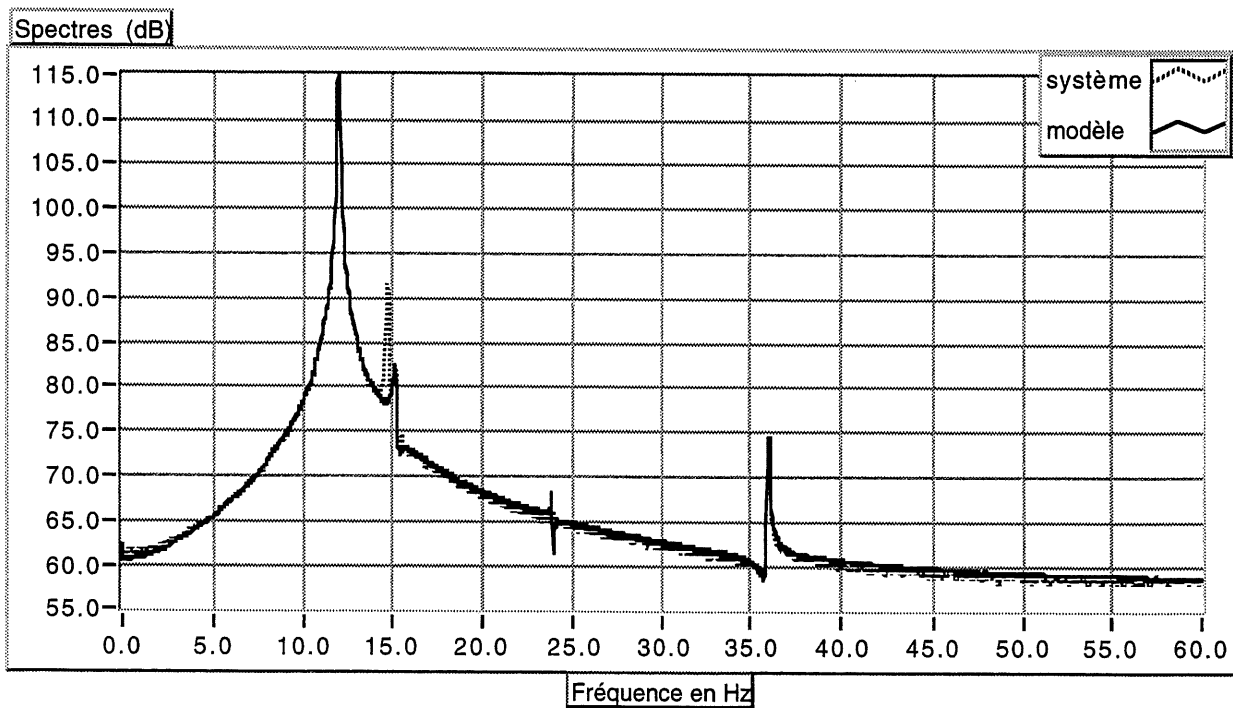


Figure 6.13(e) Vérification du modèle IV: excitation à 12.0 Hz

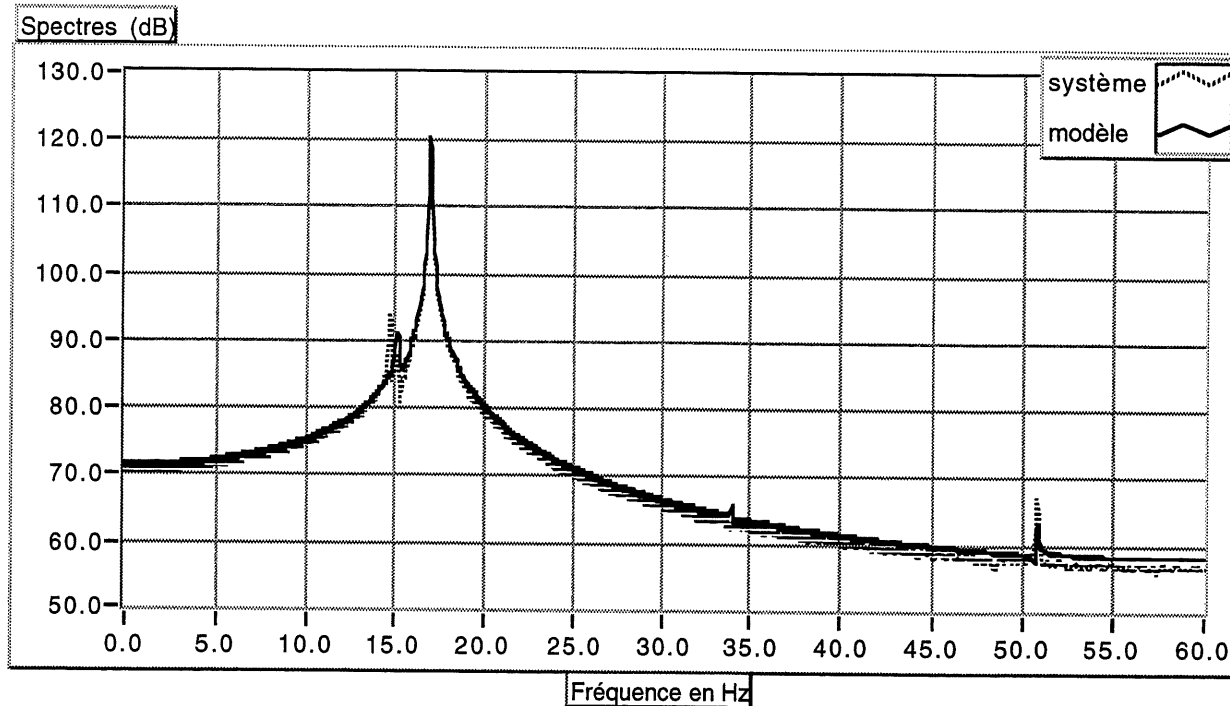


Figure 6.13(f) Vérification du modèle IV: excitation à 17.0 Hz

On peut faire quelques observations importantes sur ces résultats. Premièrement, on observe que la réponse à la fréquence fondamentale (la fréquence d'excitation), pour les deux modèles, représente très bien la réponse du système physique à la même fréquence.

Deuxièmement, on remarque que les harmoniques (dus à la non-linéarité) sont mieux représentées par le modèle IV que par le modèle III. Cela peut s'expliquer par le fait qu'on suppose une séparabilité pour le modèle III. Par contre, le modèle IV est plus général, donc on doit s'attendre à de meilleurs résultats.

Troisièmement, on observe dans les figures 6.13(b), 6.13(d) et 6.13(f) des pics à 15.1 Hz dans la réponse du modèle. On ne s'attend pas à ces pics car ils ne correspondent pas à une fréquence multiple de la fréquence d'excitation.

Ces pics sont dus au fait qu'on n'a pas attendu assez longtemps avant de prendre un bloc du signal de sortie pour calculer le spectre d'énergie. Les signaux d'excitations utilisés dans ces vérifications étaient des blocs de 2048 échantillons. Pour les 2048 échantillons du signal de sortie, on calcule le spectre d'énergie sur la deuxième moitié du signal. Cependant, à cause du fait que le système est très résonant à 15.1 Hz, la réponse transitoire est toujours discernable, même après les 1024 premières échantillons.

Finalement, on remarque qu'il y a toujours une concentration d'énergie d'une ampleur d'environ 90 dB dans la proximité de 15 Hz dans la réponse du système. Ceci est dû d'une part à la réponse transitoire du système et d'autre part au bruit (voir la section 2.9) qui est accentué à 15 Hz à cause de la résonance de la poutre.

## **6.8 Optimisation du réseau pour le modèle IV**

Dans cette section, on considère le modèle IV plus en détails car les résultats des vérifications nous montrent que c'est un meilleur modèle. On procède à une optimisation du modèle en deux étapes. Dans la première étape, on explore l'ordre temporel du système, c'est-à-dire qu'on examine l'effet de varier la taille des lignes à retard qui servent d'entrée au réseau. Cela nous permettra de réduire le nombre d'entrées pour le réseau. Dans la deuxième étape, on explore la complexité spatiale de la non-linéarité, c'est-à-dire qu'on observe l'effet de varier le nombre de neurones,  $N_1$ , dans la couche cachée pour l'ordre temporel retenu à la première étape.



On appelle  $N_x$  le nombre de sorties qui proviennent de la ligne à retard qui a pour entrée le signal  $x(n)$  et  $N_y$  pour l'autre ligne à retard. Les résultats des identifications, pour de différentes combinaisons ( $N_x$ ,  $N_y$ ), sont donnés dans le tableau 6.1. On retient l'ordre ( $N_x = 32$ ,  $N_y = 24$ ) car cette combinaison nous donne une très bonne convergence pour un minimum d'entrées du réseau. Les X représentent des combinaisons pour lesquelles l'apprentissage était impossible, même après des dizaines de milliers d'itérations. Les valeurs données sont les moyennes de la convergence en dB. L'effet de varier  $N_1$  pour  $N_x = 32$  et  $N_y = 24$  est montré dans le tableau 6.2.

TABLEAU 6.1 EFFET DE VARIER  $N_x$  ET  $N_y$  SUR LA CONVERGENCE

		$N_x$				
$N_y$		128	64	32	16	8
	32				-39	-3
	24			-40	-39	-3
	16	X	-40	X	-39	
	8	-40	-40	-39	-35	
	4	-38	-37			
	2	-35	-33			

TABLEAU 6.2 EFFET DE VARIER  $N_1$  POUR  $N_x = 32$  ET  $N_y = 24$

$N_1$	28	56	112	224
Convergence (dB)	X	-40	-41	-41

## 6.9 Commentaires sur les résultats

On remarque que dans la plupart des cas, l'apprentissage des réseaux était plutôt rapide. Aussi il semblait que l'algorithme d'adaptation n'avait pas tendance à aboutir les réseaux dans des minimums locaux.

L'identification basée sur le modèle I donne une performance à peine meilleure que le modèle ARMA linéaire. L'identification basée sur le modèle II est un peu plus performante que ce dernier, c'est-à-dire qu'on parvient à faire baisser l'énergie de l'erreur d'identification par environ 2 dB.

Les modèles III et IV sont des bien meilleurs candidats pour l'identification du système. Dans les deux cas, on parvient à faire baisser l'énergie de l'erreur par environ 6 dB (un facteur de 2 sur l'amplitude de l'erreur) par rapport à l'identification ARMA linéaire. Les deux modèles sont aussi performants si l'on juge par l'énergie de l'erreur à la fin des identifications. Cependant, lors des vérifications, on s'aperçoit que lorsque les modèles sont en boucle fermée, c'est-à-dire qu'on utilise les sorties des modèles au lieu des sorties cibles pour l'entrée des lignes à retard, l'erreur de modélisation est plus faible pour le modèle IV que pour le modèle III. En plus, les vérifications pour des excitations en régime harmonique nous montrent que le modèle IV tient mieux compte de la non-linéarité du système. Pour ces raisons, on considère que le modèle IV est le meilleur des modèles considérés.

Une optimisation du modèle IV nous montre qu'en gardant 32 échantillons en mémoire sur le signal d'entrée, et 24 échantillons en mémoire sur le signal de sortie, on obtient un bon modèle avec le minimum de mémoire. Il

paraît aussi qu'il faut environ 112 neurones dans la couche cachée, pour une meilleure représentation de la non-linéarité.

## CONCLUSION

Dans cette ouvrage, la non-linéarité de la branche de contrôle d'un système de contrôle de vibrations muni d'un actuateur piézo-électrique et d'un capteur PVDF a été évaluée et des identifications de la branche de contrôle par des modèles basés sur les réseaux de neurones ont été considérées.

En excitant le système par des harmoniques, on a pu démontrer que la non-linéarité du système est assez importante. Une identification basée sur un modèle ARMA linéaire nous donnait une convergence dans les alentours de -34 dB. En faisant des identifications avec certains modèles non-linéaires basés sur des réseaux de neurones, on a pu baisser la convergence jusqu'aux alentours de -41 dB. Cela nous offre une bien meilleure représentation du système. Le modèle retenu a la forme suivante :

$$y(n) = f[x(n), x(n-1), x(n-2), \dots, x(n-31), y(n-1), y(n-2), \dots, y(n-24)]$$

où  $x(n)$  est le signal d'entrée,  $y(n)$  est la sortie et  $f$  est une fonction non-linéaire.

Les problèmes souvent rencontrés dans l'apprentissage des réseaux de neurones sont la faible vitesse d'apprentissage et la tendance à tomber dans des minimums locaux. Par contre, le type de réseau utilisé dans ce travail et l'algorithme d'apprentissage pour ce réseau donnaient, en général, des convergences assez rapides et semblaient éviter les minimums locaux.

On a vu que lors des identifications, la présence du bruit dans l'environnement biaise les résultats des identifications, surtout pour les identifications des modèles récurrents en mode série-parallèle. Il est important dans

l'avenir de faire des recherches sur l'identification des systèmes non-linéaires par des modèles récurrents, en la présence du bruit.

Avec le succès de ces recherches plusieurs avenues sont ouvertes vers des travaux futurs. En conséquence, on pourrait considérer le problème de développement des contrôleurs non-linéaires pouvant être basés sur le type de réseau de neurones considéré ici, en utilisant le modèle identifié de la branche de contrôle. On doit aussi remarquer que les techniques d'identification présentées dans ce travail peuvent aussi être utilisées dans les problèmes de contrôle du bruit avec les haut-parleurs, par exemple pour la réduction de bruit dans des conduits ou cheminées, puisque la non-linéarité de ce type d'actuateur est un problème bien connu qui limite la performance d'un contrôle.

## ANNEXE 1

### MODES DE VIBRATION POUR UNE POUTRE UNIFORME ENCASTRÉE-LIBRE

Si  $h$  = épaisseur  
 $L$  = longueur  
 $E$  = module de Young  
 $\rho$  = densité du matériel

alors les fréquences propres de vibrations sont données par:

$$f_n = \frac{\lambda_n^2}{2\pi L^2} \sqrt{\frac{Eh^2}{12\rho}}$$

où  $\lambda_1 = 1.87510407$   
 $\lambda_2 = 4.69409113$   
 $\lambda_3 = 7.85475744$   
 $\lambda_4 = 10.99554073$

et les modes de vibrations sont données par:

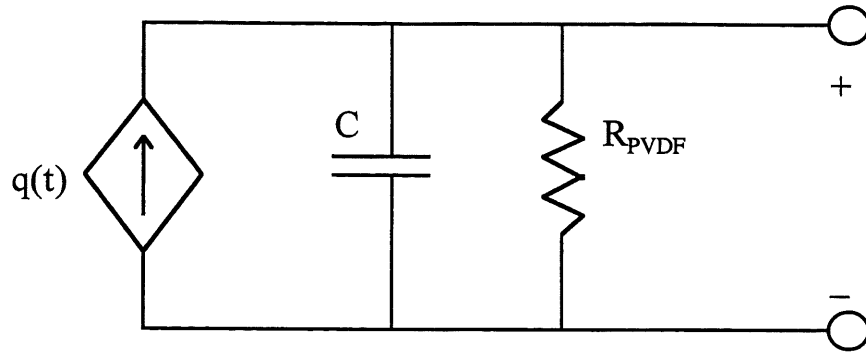
$$\cosh\left(\frac{\lambda_n x}{L}\right) - \cos\left(\frac{\lambda_n x}{L}\right) - \sigma_n \left[ \sinh\left(\frac{\lambda_n x}{L}\right) - \sin\left(\frac{\lambda_n x}{L}\right) \right]$$

où  $\sigma_1 = 0.734095514$   
 $\sigma_2 = 1.018467319$   
 $\sigma_3 = 0.999224497$   
 $\sigma_4 = 1.000033553$

## ANNEXE 2

### MODÉLISATION D'UN CAPTEUR PVDF

Le circuit équivalent pour un capteur PVDF est donné ci-dessous.



La charge  $q(t)$  est proportionnelle à la déformation en longueur.

Lorsque le capteur est connecté à un préamplificateur avec impédance d'entrée  $R_{in}$  et gain  $K$ , la fonction de transfert entre la sortie du préamplificateur et la charge sur le capteur est donnée par:

$$\frac{V(s)}{Q(s)} = \frac{Ks / C}{s + 1/CR}$$

où  $R$  est la résistance combinée de  $R_{PVDF}$  et  $R_{in}$  en parallèle.

En supposant que la déformation en longueur est proportionnelle à la déflexion latérale,  $w(t)$ , de la poutre alors:

$$\frac{V(s)}{W(s)} \propto \frac{s}{s + 1/CR}$$

## BIBLIOGRAPHIE

- [1] BOZICH, D.J., MACKAY, H.B., *Neurocontrollers Applied to Real-Time Vibration Cancellation at Multiple Locations*, Science Applications International Corporation, 4161 Campus Point Court, San Diego, CA 92121, U.S.A..
- [2] BOSE, N.K., LIANG, P. (1996) *Neural Network Fundamentals with Graphs, Algorithms, and Applications*, MacGraw-Hill
- [3] CHEUNG, W-S. (mars 1993) *Identification, Stabilisation and Control of Nonlinear Systems using Neural Network-Based Parametric Nonlinear Modelling*, University of Southampton, Thèse de doctorat, Faculty of Engineering and Applied Science, 158 p.
- [4] CHUNG WON, C., SULLA, J.L. (1994) *Experiments on the Linearization of a Piezoelectric Structure*, Proc. SPIE - Int. Soc. Opt. Eng., U.S.A., vol. 2192, p. 168-175.
- [5] HAYKIN, S. (1996) *Adaptive Filter Theory*, Prentice Hall, New Jersey, 3<sup>e</sup> édition, Thomas Kailath, 989 p.
- [6] HAYKIN, S. (mars 1996) *Neural Networks Expand SP's Horizons*, IEEE Signal Processing Magazine, vol. 13, n° 2, p. 24-49.
- [7] HORNIK, K., STINCHCOMBE, M., WHITE, H. (1989) *Multilayer Feedforward Networks are Universal Approximators*, Neural Networks, Pergamon Press Ltd., Oxford, England, vol. 2, p. 359-366.
- [8] LE DINH, C.T. (1995) *Notes de Cours GEI 752 - Traitements avancés des signaux*, Département de génie électrique et génie informatique, Faculté des sciences appliquées, Université de Sherbrooke.
- [9] LOW, S., HAWKSFORD, M.O.J. (1993) *A Neural Network Approach to the Adaptive Correction of Loudspeaker Nonlinearities*, Audio Engineering Society.
- [10] NARENDRA, K.S., PARTHASARATHY, K. (mars 1990) *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Trans. on Neural Networks, vol. 1, n° 1, p. 4-27.



- [11] NOBAKHT, R.A., ARDALAN, S.H. (1993) *Modelling and Adaptive Filtering of Systems with Hysteresis Nonlinearities* (June 1992) SUPERCOMM/ICC '92, Discovering a New World of Communications, Chicago, IL, USA, vol. 2, p. 813-17.
- [12] NORTON, J.P. (1986) *An Introduction to Identification*, London, Academic Press, Harcourt Brace Jovanovich Publishers, 310 p.
- [13] PITAS, I., VENETSANOPOULOS, A.N. (1990) *Nonlinear Digital Filters*, Kluwer Academic Publishers, Boston, U.S.A., 387 p.
- [14] RAO, S.S. (1995) *Mechanical Vibrations*, Addison-Wesley Publishing Company, 3<sup>e</sup> édition, Johnson, S., 912 p.
- [15] SCALERO, R.S., TEPEDELENLIOGLU, N. (janvier 1992) *A Fast New Algorithm for Training Feedforward Neural Networks*, IEEE Transactions on Signal Processing, vol. 40, No. 1, p. 202-210.
- [16] SCHOUKENS, J., PINTELON, R. (1991) *Identification of Linear Systems*, Pergamon Press, Oxford, England (s.éd), 332 p.
- [17] SHANMUGAM, K.S., LAL, M. (janvier 1976) *Analysis and Synthesis of a Class of Nonlinear Systems*, IEEE Transactions on Circuits and Systems, vol. CAS-23, No. 1, p.17-25.
- [18] SHANMUGAM, K.S. (1979) *Digital and Analog Communication Systems*, John Wiley & Sons
- [19] SHYNK, J.J. (avril 1989) *Adaptive IIR Filtering*, IEEE ASSP Magazine, vol. 6, No. 2, p. 4-21.
- [20] TSIMBINOS, J., LEVER, K.V. (1994) *Sampling Frequency Requirements for Identification and Compensation of Nonlinear Systems*, IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 3, p.III/513-16.

- [21] SAATY, T.L., BRAM, J. (1964) *Nonlinear Mathematics*, McGraw-Hill, états unis, W.T. Martin et E.H. Spanier, 381 p.